



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

TIMO RUOSTILA

MOBIILILAITTEIDEN HUOMIOIMINEN WEB-POHJAISEN CRM- JÄRJESTELMÄN JATKOKEHITYKSESSÄ

Diplomityö

Tarkastajat: Heli Vääätäjä, Satu
Jumisko-Pyykkö

Tarkastaja ja aihe hyväksytty Tieto- ja
sähkötekniikan tiedekuntaneuvoston
kokouksessa 18. elokuuta 2014

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

RUOSTILA, TIMO: Mobiililaitteiden huomioiminen web-pohjaisen CRM-järjestelmän jatkokehityksessä

Diplomityö, 51 sivua, 4 liitesivua

Lokakuu 2014

Pääaine: Käytettävyys

Tarkastajat: Heli Väättäjä, Satu Jumisko-Pyykkö

Avainsanat: Käyttöliittymäsuunnittelu, käytettävyys, käyttäjätutkimus, responsiivinen suunnittelu, mobiililaitte, mobiilisovellus, hybridisovellus

Mobiililaitteiden huomioimisesta web-järjestelmissä on tullut entistä tärkeämpää, koska mobiililaitteiden käytön määrä kasvaa jatkuvasti ja ne ovat muuttaneet merkittävästi tapaa, jolla käytämme Internetiä. Tämän työn tarkoituksena on vertailla eri lähestymistapoja mobiililaitteiden huomioimiselle ja toteuttaa olemassa olevaan CRM-järjestelmään (Customer Relationship Management) prototyyppi käyttöliittymästä käyttäen tilanteeseen sopivaa lähestymistapaa.

Työ koostuu teoriaosuudesta ja empiirisestä osuudesta, jossa toteutetaan kolmivaiheinen käyttäjätutkimus. Teoriaosuudessa tehdään aluksi kartoitus eri lähestymistavoista mobiililaitteiden huomioimiselle web-järjestelmissä. Korkeimmalla tasolla vaihtoehtoina on luoda erillinen mobiilisovellus tai toteuttaa järjestelmän käyttöliittymä hyödyntäen niin sanottua responsiivista web-suunnittelua (Responsive Web Design). Responsiivisen web-suunnittelun avulla järjestelmän käyttöliittymä voidaan toteuttaa siten, että se mukautuu käytettävän laitteen näytön kokoon. Lisäksi teoriaosuudessa tehdään kirjallisuuskatsaus käyttäjäkeskeisiin menetelmiin ja prosesseihin sekä CRM-järjestelmiin.

Käyttäjätutkimuksen ensimmäisessä vaiheessa tutkittiin käytössä olevan CRM-järjestelmän käyttäjien tarpeita erityisesti mobiililaitteilla käytettäessä. Kerätyn tiedon pohjalta luotiin matalan tason prototyyppi, jota testattiin käyttäjillä tutkimuksen toisessa vaiheessa. Toisen vaiheen tarkoituksena oli varmistaa suunnitteluideoiden toimivuus, minkä jälkeen prototyypistä luotiin korkean tason versio hyödyntäen responsiivista web-suunnittelua. Tutkimuksen kolmannessa vaiheessa korkean tason prototyyppiä testattiin rinnakkain käytössä olevan järjestelmän kanssa ja näiden tuloksia verrattiin toisiinsa.

Opinnäytetyössä tehty tutkimus osoitti, että mobiililaitteita käytetään pääasiassa tiedon tarkistamiseen. Lisäksi tuloksien perusteella prototyyppi sai käyttäjiltä myönteisen vastaanoton ja responsiivisen web-suunnittelun avulla saatiin tilastollisesti merkittäviä positiivisia vaikutuksia kyseisen järjestelmän käytettävyyteen.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

RUOSTILA, TIMO: Considering mobile devices in further development of a web-based CRM system

Master of Science Thesis, 51 pages, 4 Appendix pages

October 2014

Major: Usability

Examiners: Heli Väättäjä, Satu Jumisko-Pyykkö

Keywords: User Interface Design, Usability, User testing, Responsive design, Mobile device, Mobile application, Hybrid application

Considering mobile devices in web-based systems has become important because of the increasing usage of mobile devices which have radically changed the way we use the Internet. The purpose of this thesis is to find out what different approaches there are to consider mobile devices better and to implement user interface prototype for an existing CRM-system (Customer Relationship Management) using a suitable approach.

The thesis consists of a theory part and an empiric part where a three phase user study was conducted. In the theory part the different approaches for considering mobile devices in web-based systems are explored and compared. In the highest level the main approaches are implementing a separate mobile application and implementing the user interface of the system using so called responsive web design. With responsive web design it is possible to design user interfaces that adapt to the screen of the device that is being used. In addition in the theory part a literature review is done in user-centered design processes and methods and in CRM-systems.

In the first phase of the user study the needs for the existing CRM-system were studied especially when using it with mobile devices. Based on the information gathered a low level prototype was implemented and tested in the second phase of the study. The idea of the second phase was to validate the design ideas after which a high level prototype was implemented using responsive web-design. In the third phase of the study the high level prototype was tested side by side with the existing system and the results were compared to each other.

The study conducted in this thesis showed that mobile devices are mostly used to check information. Based on the results of the study the prototype got a positive reception from the users and statistically significant improvements were achieved to the usability of the system by using responsive web design.

ALKUSANAT

Tämä opinnäytetyö tehtiin Leanware Oy:lle asiakasprojektiin liittyen. Olen kiitollinen Leanwarelle mahdollisuudesta käytännönläheisen ja mielenkiintoisen diplomityön tekemiseen.

Vaikka työ oli suhteellisen suppea ja käytännönläheinen, työn toteutukseen liittyi useita haasteita. Haluan ilmaista kiitokset esimiestäni Juha Kihlakaskea kohtaan näiden haasteiden kanssa auttamisesta sekä ylipäättään koko aiheen ja työn mahdollistamisesta. Iso kiitos menee myös työn kirjoitusprosessin aikana paljon tukea antaneille ystäville, työkavereille ja perheenjäsenille.

Kiitän myös ohjaajiani Heli Väätäjää ja Satu Jumisko-Pyykköä joustavasta työskentelystään diplomityön parissa sekä konkreettisista kommentteista ja rakentavasta tavasta antaa palautetta.

Tampere, 14. lokakuuta, 2014.

Timo Ruostila

SISÄLLYSLUETTELO

1.	JOHDANTO	1
2.	MOBIILILAITTEIDEN HUOMIOIMINEN JA TEKNOLOGIAT	3
2.1	Mobiilisovellukset.....	4
2.1.1	Natiivisovellukset.....	4
2.1.2	Mobiilit HTML5 web-sovellukset	5
2.1.3	Hybridisovellukset	6
2.2	Responsiivinen web-suunnittelu	7
2.2.1	Teknologiat ja toteutus.....	8
2.2.2	Huomioita.....	10
2.3	Lähestymistapojen vertailu	11
3.	KÄYTTÄJÄKESKEINEN SUUNNITTELU JA PROSESSIT	15
3.1	Käyttäjakeskeinen suunnittelu	15
3.2	Mobiilikäyttöliittymien suunnitteluperiaatteet	16
3.3	Tutkimusmenetelmät.....	18
3.3.1	Contextual Inquiry	18
3.3.2	Prototypointi.....	20
4.	PROTOTYYPIN SUUNNITTELUPROSESSI.....	22
4.1	CRM-järjestelmät.....	22
4.2	Tapaus: web-pohjainen CRM-järjestelmä.....	22
4.3	Prototyypin toteutus	23
5.	VAIHE 1 – TARPEIDEN KARTOITUS	25
5.1	Menetelmä.....	25
5.2	Tulokset.....	27
6.	VAIHE 2 – PROTOTYYPIN ITEROINTI.....	31
6.1	Menetelmä.....	31
6.2	Tulokset.....	32
6.3	Prototypointi.....	33
7.	VAIHE 3 – KÄYTETTÄVYYSTESTAUS	36
7.1	Menetelmä.....	36
7.2	Tulokset.....	39
8.	JOHTOPÄÄTÖKSET	43
9.	LÄHTEET.....	46

LIITE A: KÄYTETTÄVYYSTESTAUKSEN KYSELY

LIITE B: MITATUT SUORITUSAJAT

TERMIT JA NIIDEN MÄÄRITELMÄT

Contextual Inquiry	Järjestelmän käyttökontekstin ja käyttäjien tarpeiden tutkimiseen tarkoitettu tutkimusmenetelmä, joka on osa isompaa tutkimusmenetelmäkokonaisuutta nimeltä Contextual Design.
CRM-järjestelmä	CRM-järjestelmät (Customer Relationship Management) ovat järjestelmiä, joiden avulla tehostetaan asiakkaisiin liittyvän tiedon hyödyntämistä myynissä, johtamisessa ja markkinoinnissa.
Hybridisovellus	Hybridisovellukset ovat mobiileja web-sovelluksia, jotka on paketoitu natiiveiksi sovelluksiksi hyödyntäen tähän jotain tarkoitettua ohjelmistokehystä (esimerkiksi PhoneGap).
Responsiivinen web-suunnittelu	Responsiivinen web-suunnittelu on uusista web-teknologioista koottu kokonaisuus, jonka avulla voidaan tehdä web-järjestelmän käyttöliittymä siten, että se mukautuu käytössä olevan laitteen näytön kokoon.
Natiivisovellus	Käyttöjärjestelmäkohtainen mobiilisovellus, jota ajetaan laitteessa.
Mobiili HTML5 web-sovellus	Täysin web-pohjainen mobiilisovellus, jota ajetaan käyttöjärjestelmän mobiiliselaimessa.
Polyfill	Internet-selaimen puutteellisia ominaisuuksia simuloiva usein JavaScript-ohjelmointikielellä toteutettu ratkaisu.
Responsiivinen web-suunnittelu	Responsiivinen web-suunnittelu on uusista web-teknologioista koottu kokonaisuus, jonka avulla voidaan tehdä web-järjestelmän käyttöliittymä siten, että se mukautuu käytössä olevan laitteen näytön kokoon.

1. JOHDANTO

Teknologian kehitys ja mobiililaitteiden käytön lisääntyminen muuttaa jatkuvasti tapaa jolla käytämme Internetiä. Vuonna 2003 Microsoft hallitsi Internetin käyttöä: 95 % Internetin käytöstä tehtiin Internet Explorer selaimella työpöytätietokoneilla Windows-käyttöjärjestelmän kautta. Nykyisin on olemassa useita yleisesti käytössä olevia Internet selaimia ja käyttöjärjestelmiä ja uusia tulee lisää jatkuvasti. Lisäksi suuri osa Internetin käytöstä tehdään erilaisten mobiililaitteiden kautta: arvioiden mukaan Internetin käyttö mobiililaitteilla on ylittänyt ajallisesti Internetin käytön työpöytäselaimilla. Tämä kehitys tekee mobiililaitteiden huomioimisesta entistä tärkeämpää, mikä luo uusia tarpeita käyttäjille ja asettaa haasteita sovelluksien kehittäjille.

Parhaillaan käydään jatkuvaa vertailua, mikä lähestymistapa mobiililaitteiden huomioimiseen web-järjestelmissä on paras nyt ja tulevaisuudessa. Yleisesti katsotaan, että vaihtoehtoina on toteuttaa erillinen mobiilisovellus tai kaikille laitteille yhtenäinen responsiivinen käyttöliittymä. Molemmilla vaihtoehdoilla on hyvät ja huonot puolensa. Mobiilisovelluksen olisi hyvä toimia usealla eri alustalla, mikä tekee natiivisovelluksien toteuttamisesta todella kallista. Hybridisovellukset toimivat puolestaan kaikilla alustoilla, mutta eivät vastaa käyttäjäkokemukseltaan natiivisovelluksia. Web-pohjaiset sovellukset ovat myös laiteriippumattomia ja hybridisovelluksiin verrattuna halvempia toteuttaa, mutta eivät toistaiseksi mahdollista laiteominaisuuksien kuten erilaisten sensoreiden hyödyntämistä.

Responsiivinen web-suunnittelu on uusista web-teknologioista koottu kokonaisuus, jonka avulla voidaan tehdä web-järjestelmän käyttöliittymä siten, että se mukautuu käytössä olevan laitteen näytön kokoon. Tällä lähestymistavalla on merkittäviä etuja, kuten parempi ylläpidettävyyys ja kustannustehokkuus muihin vaihtoehtoihin verrattuna. Responsiivisen web-suunnittelun avulla saavutetaan yhtenäinen ulkoasu kaikille laitteille ja alustoille, mutta sen haasteena on tällä hetkellä suorituskyky.

Tiedonhallinta ja tiedolla johtaminen on kasvanut merkittäväksi kilpailuvaltiksi nykypäivänä yritysmaailmassa. Tästä syystä CRM-järjestelmien (Customer Relationship Management) käyttö on kasvattanut suosiotaan viime vuosikymmenillä. Ne ovat järjestelmiä, joiden avulla tehostetaan asiakkaisiin liittyvän tiedon hyödyntämistä myynnissä, johtamisessa ja markkinoinnissa.

Tässä työssä kehitetään asiakkaan CRM-järjestelmää siten, että se huomioi mobiililaitteet entistä paremmin. Opinnäytetyön tutkimuskysymys on ”Kuinka mobiililaitteet voidaan huomioida web-pohjaisen CRM-järjestelmän

jatkokehityksessä?” Teoriaosuudessa tehtävän mobiililaitteiden lähestymistapojen vertailun ja huomioon otettavien asioiden kartoituksen pohjalta valitaan lähestymistapa, jolla mobiililaitteiden huomiointi toteutetaan. Kehitettävästä järjestelmästä tehdään käyttäjäkeskeistä suunnittelua hyödyntäen prototyyppi soveltaen valittua lähestymistapaa.

Työ jakaantuu teoriaosuuteen ja empiiriseen osuuteen siten, että johdantoluvun jälkeen luvut 2-4 ovat teoriaosuutta ja luvut 5-7 empiiristä osuutta. Ensimmäisessä teorialuvussa käydään läpi mobiililaitteiden huomioimiseen web-järjestelmissä liittyvät tekijät ja lähestymistavat ja toisessa luvussa tehdään kirjallisuuskatsaus käyttäjäkeskeiseen suunnitteluun ja prosesseihin. Kolmas teorialuku kertoo CRM-järjestelmistä yleensä, kehitettävästä järjestelmästä sekä työssä tehtävän prototyypin toteutustavasta.

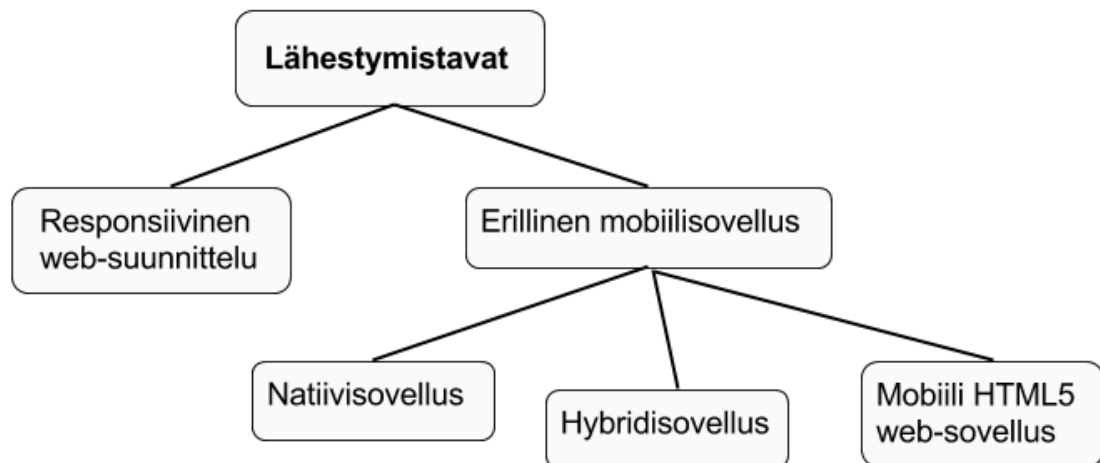
Työn empiirisessä osuudessa tehdään kolmivaiheinen käyttäjätutkimus, jonka ensimmäisessä vaiheessa kartoitetaan käytössä olevan järjestelmän käyttäjien tarpeita ja tutkitaan järjestelmän käyttöä erityisesti mobiililaitteilla käytettäessä. Kerätyn tiedon perusteella luodaan ensiksi paperiprototyyppiä hyödyntäen matalan tason prototyyppi, jota testataan tutkimuksen toisessa vaiheessa. Toisen vaiheen tutkimuksen pohjalta prototyyppiä iteroidaan ja luodaan vielä korkean tason prototyyppi. Tutkimuksen kolmannessa vaiheessa tehdään käyttäjätutkimus, jossa korkean tason prototyypin ja käytössä olevan järjestelmän käyttöä testataan ja verrataan toisiinsa.

Empiirisen osuuden jälkeen työn tulokset ja johtopäätökset kerätään yhteen luvussa 8. Johtopäätöksissä arvioidaan työssä tehtyjä ratkaisuja ja tuloksia sekä keskustellaan ehdotuksista työlle opinnäytetyön jälkeen.

2. MOBIILILAITTEIDEN HUOMIOIMINEN JA TEKNOLOGIAT

Tässä luvussa pohditaan mitä vaihtoehtoisia lähestymistapoja mobiililaitteiden huomioimiselle on olemassa (Kuva 1) ja vertaillaan niitä. Luvussa tehdään kartoitus vaihtoehtojen heikkouksille ja vahvuuksille toisiinsa verrattuna, jotta pystytään valitsemaan tässä työssä tehtävälle prototyypille soveltuvin vaihtoehto.

Mobiililaitteiden huomioiminen web-järjestelmissä



Kuva 1: Mobiililaitteiden huomioimisen lähestymistavat web-järjestelmissä.

Laitemarkkinoiden nopea kehitys tekee mobiililaitteiden huomioimisesta ajankohtaisen aiheen. Maailmanlaajuisten arvioiden [13] [22] mukaan älypuhelimien ja tablettien määrä laitemarkkinoilla suhteessa pöytäkoneisiin ja kannettaviin tietokoneisiin tulee jatkamaan kasvuaan. Gartnerin tekemän tutkimuksen [13] mukaan matkapuhelinten toimituksille arvioidaan 3,1 prosentin kasvua 1,9 miljardiin toimitettuun laitteeseen vuonna 2014 edelliseen vuoteen verrattuna. Tutkimuksessa arvioitiin 66 % myydyistä matkapuhelimista olevan älypuhelimia vuonna 2014, mutta tämän luvun ennustetaan nousevan 88 prosenttiin vuoteen 2018 mennessä. Internetin käyttö mobiililaitteiden avulla tulee siis kasvamaan tulevaisuudessa, joten niiden huomioimisesta tulee entistäkin tärkeämpää.

Taulukossa 1 on esitetty International Data Corporationin (IDC) tutkimuksen [22] ennuste vuosille 2013 - 2017 laitemarkkinoiden kehityksestä. Sen mukaan laitemarkkinoiden arvioidaan kasvavan kokonaisuudessaan 58,1 prosenttia samalla kun

pöytätietokoneiden toimituksien määrän arvioidaan kääntyvän laskuun ja kannettavien tietokoneiden toimitusten määrän kasvun hidastuvan. Tutkimuksessa arvioitiin tablettien toimitusten määrän kasvavan eniten tällä aikavälillä ja niiden arvioitiin ylittävän kaikkien PC-laitteiden toimitukset vuoden 2015 loppuun mennessä [22]. Arvio on osoitus siitä, että Internetin käyttö tulee tapahtumaan tulevaisuudessa yhä vähemmän pöytätietokoneiden ja kannettavien tietokoneiden avulla ja sen sijaan entistä laajemmalla skaalalla erilaisia laitteita.

Taulukko 1: Ennuste laitemarkkinoiden kehitykselle vuosina 2013 - 2017 (toimituksia miljoonissa) [22].

Tuote kategoria	Toimitukset 2013	2013 Markkinaosuus	Toimitukset 2017	2017 Markkinaosuus	2013 - 2017 Kasvu
Pöytäkoneet	134,4	8,6 %	123,11	5 %	-8,4 %
Kannettavat	180,9	11,6 %	196,6	8 %	8,7 %
Tabletit	227,3	14,6 %	406,8	16,5 %	78,9 %
Älypuhelimet	1013,2	65,1 %	1733,9	70,5 %	71,1 %
Yhteensä	1556	100 %	2460,5	100 %	58,1 %

Tutkimuksen [11] mukaan 83 % ihmisistä pitää saumatonta kokemusta eri laitteiden välillä tärkeänä. Tällä hetkellä yritykset eivät huomioi mobiililaitteita web-sovelluksissa optimaalisella tavalla: 54 % ihmisistä on tyytymättömiä yritysten tapaan tuoda sisältöä mobiililaitteille [52]. Mobiililaitteet täytyy huomioida joko erillisellä mobiilisovelluksella tai kaikille laitteille yhtenäisellä responsiivisella sivustolla, joka muuttaa ulkoasuun sen mukaan millä laitteella sitä käytetään [ibid.].

2.1 Mobiilisovellukset

On olemassa useita lähestymistapoja toteuttaa mobiilisovelluksia. Yleisimmin listataan kolme vaihtoehtoa: natiivisovellukset, mobiilit HTML5 web-sovellukset sekä hybridisovellukset. Joissain lähteissä (esimerkiksi [21]) web-pohjaiset mobiilisovellukset jaotellaan lisäksi HTML5-ratkaisujen lisäksi perinteisiin mobiilisivustoihin. Tämän työn kannalta tämä jaottelu ei ole kuitenkaan oleellista, koska mobiilien HTML5 web-sovelluksien ja mobiilisivujen väliset erot eivät ole muihin lähestymistapoihin verrattuna merkittävän suuria. Parhaillaan käydään jatkuvaa keskustelua siitä mikä näistä lähestymistavoista on paras ja mikä tulee vallitsemaan tulevaisuudessa.

2.1.1 Natiivisovellukset

Käyttöjärjestelmiä kehittävät yritykset haluavat, että laitteille tehtävät sovellukset ovat käyttöjärjestelmälle ominaisia ja hyödyntävät käyttöjärjestelmän tarjoamia ominaisuuksia mahdollisimman hyvin [48]. Natiivisovellukset ovat

käyttöjärjestelmäkohtaisia sovelluksia, joiden avulla tämä on mahdollista. Natiivisovelluksia toteuttaessa on huomioitava, että jokaiselle käyttöjärjestelmälle, esimerkiksi iOS, Android, Windows, jolla sovelluksen tulee toimia, täytyy ohjelmoida oma sovellus ja ylläpitää sitä. Vaikka sovellus olisi pohjimmiltaan web-sovellus, sille voidaan tehdä natiivisovellus mobiililaitteita varten, joka keskustelee tätä varten kehitetyn web-ohjelmointirajapinnan kanssa tiedonvälitystä varten.

Natiivisovelluksien avulla on toistaiseksi mahdollista luoda parempi käyttäjäkokemus muihin mobiilisovelluksien toteuttamisen lähestymistapoihin verrattuna. Tämä johtuu osittain siitä, että natiivisovellukset hyödyntävät käyttöjärjestelmän omia rajapintoja ja abstraktiotasoja grafiikan piirtämiseen näytölle. Web-teknologioiden HTML ja CSS koodin tulkkaukseen verrattuna käyttöjärjestelmän natiivisovellukset ovat suorituskyvyltään tehokkaampia, mikä parantaa sovelluksen käyttäjäkokemusta. Lisäksi natiivisovellukset mahdollistavat laitteelle ominaisten toimintojen eli laiterajapinnan (Device API) hyödyntämisen. Laiterajapinnan kautta sovelluksessa voidaan hyödyntää esimerkiksi kameraa ja erilaisia antureita, joiden hyödyntäminen ei toistaiseksi onnistu täysin mobiiliselaimissa ajettavissa sovelluksissa [48]. Erilaisien antureiden ja ilmoitusten käyttäminen voi usein olla elintärkeää sovelluksen käyttäjäkokemuksen kannalta [8].

Natiivisovellukset kirjoitetaan yleisimmin jollain C-pohjaisella ohjelmointikielellä tai Javalla, jotka ovat käännettäviä ohjelmointikieliä. Esimerkiksi Applen iOS käyttöjärjestelmän sovelluksien ohjelmoiminen vaatii C ja Objective C kielten osaamista, Googlen Android käyttöjärjestelmä vastaavasti Java ohjelmointikielen osaamista ja Microsoftin Windows käyttöjärjestelmä C# ohjelmointikielen .NET ohjelmointikehyksen osaamista. Kaikilla alustoilla on ohjelmointikielen lisäksi omat kehitystyökalut, kääntämiseen liittyvät käytännöt ja ohjelmointirajapinnat. [8]

2.1.2 Mobiilit HTML5 web-sovellukset

Mobiilit HTML5 web-sovellukset ovat pohjimmiltaan normaaleja web-sivustoja, jotka on suunniteltu pienellä näytöllä käytettäväksi. HTML5 standardin ansiosta selaimessa ajettavan sovelluksen kokemuksesta saadaan enemmän natiivin sovelluksen kaltainen perinteiseen nettisivuun verrattuna. Mobiilit web-sovellukset ohjelmoidaan perinteisten web-ohjelmointikielten avulla (HTML, CSS, JavaScript) ja sovellusta ajetaan laitteen selaimessa. Tästä syystä mobiilit web-sovellukset ovat alustariippumattomia eli niitä voidaan ajaa millä tahansa modernilla mobiiliselaimella riippumatta käytössä olevasta käyttöjärjestelmästä. Niitä ei lisäksi tarvitse asentaa erikseen eikä sovelluksen levitykseen tarvita erityistä sovelluskauppaa, vaan ne voidaan avata navigoimalla oikeaan URL osoitteeseen selaimessa. [28]

HTML5 standardin kehitys on tuonut mukanaan joukon hyödyllisiä työkaluja ja ohjelmointikehyksiä, joiden avulla on mahdollista kehittää kauniita käyttöliittymiä

todella nopeasti. Web-teknologioiden haittapuoli on kuitenkin se, että web-alustasta ei ole täyttä pääsyä laitteiden alemman tason rajapinnoille, joilla pystytään hyödyntämään laitteiden omia toimintoja, kuten ilmoituksia ja antureita. Tätä haittapuolta ollaan kuitenkin korjaamassa ja nykyisin monet mobiiliselaimet tarjoavatkin jo joitain alemman tason rajapintoja, joita voidaan hyödyntää suoraan web-teknologioiden avulla. [8] Mobiiliselaimien pääsystä laiterajapintaan vastaa WWW-standardeja kehittävän W3C (World Wide Web Consortium) organisaation työryhmä [10].

Mobiilien web-sovelluksien käyttäjäkokemus kärsii laiterajapinnan puutteellisen pääsyn lisäksi siitä, että niiden käyttöliittymä luodaan tulkitsemalla HTML, CSS ja JavaScript koodia. Tämä on merkittävästi hitaampaa natiivisovelluksien suorituskykyyn verrattuna ja aiheuttaa lisäksi eroavaisuuksia eri mobiiliselaimien välillä [8].

2.1.3 Hybridisovellukset

Hybridisovellukset ovat mobiileja web-sovelluksia, jotka on paketoitu natiiveiksi sovelluksiksi [48]. Kaikissa mobiilikäyttöjärjestelmissä on mobiiliselain, jonka JavaScript liittymän kanssa pystytään kommunikoimaan ohjelmallisesti natiivikoodista. Tätä ideaa hyödyntämällä Eric Oesterle, Rob Ellis ja Brock Whitten keksivät PhoneGap:in, joka on suosituin esimerkki hybridisovellusten toteuttamisen mahdollistavasta ohjelmistokehyksestä. PhoneGap on avoimen lähdekoodin ohjelmistokehys, joka syntyi vuonna 2008 iPhoneDevCampissa ja se sovitettiin myöhemmin myös muille alustoille.

PhoneGapin kaltaiset ohjelmistokehykset tarjoavat kehittäjille ympäristön, jossa he voivat toteuttaa sovelluksia web-teknologioilla (HTML, CSS, JavaScript) ja silti hyödyntää natiiveja laiteominaisuuksia ja sensoreita kutsumalla niitä kaikille mobiiliselaimille yhteisen JavaScript rajapinnan avulla [8]. Hybridisovelluksissa valitut osat sovelluksesta on siis toteutettu käyttäen web-teknologioita [28]. Web-sovellusosa voidaan joko paketoita binääriin sisään tai ajaa palvelimella, mutta mobiilisovellusta ajetaan joka tapauksessa lokaalisti käyttöjärjestelmässä hyvin samaan tapaan kuin natiivisovellustakin [ibid.].

Hybridisovellus-lähestymistavan avulla on mahdollista luoda kaikilla käyttöjärjestelmillä toimivia ja yhtenäisiä eli alustariippumattomia sovelluksia, joita voidaan jakaa alustojen omissa sovelluskaupoissa. Niiden suorituskyky ei kuitenkaan vastaa aivan natiivisovelluksien suorituskykyä, mikä vaikuttaa negatiivisesti käyttäjäkokemukseen. Mobiileihin HTML5 web-sovelluksiin verrattuna käyttäjäkokemusta voidaan optimoida kuitenkin paljon paremmin esimerkiksi laiterajapinnan tarjoamien mahdollisuuksien avulla. [28]

2.2 Responsiivinen web-suunnittelu

Responsiivinen web-suunnittelu (Responsive Web Design, RWD) on erillisen mobiilisovelluksen toteuttamisen rinnalla toinen tapa huomioida mobiililaitteet web-pohjaisessa järjestelmässä. Responsiivisen web-suunnittelun avulla voidaan luoda käyttöliittymä siten, että se mukautuu käytössä olevan laitteen näytön kokoon. On tilanteita, jolloin erillisen käyttäjäkokemuksen luominen mobiililaitteita varten voi olla järkevää, mutta viime vuosien teknologian kehitys on osoittanut, että erilaisten laitteiden määrän kasvaessa se on kuitenkin usein mahdotonta rajallisten resurssien puitteissa. Responsiivinen web-suunnittelu on tarkoitettu vastaukseksi tähän ongelmaan. [31, s.6]

Responsiivinen web-suunnittelu on uusi lähestymistapa nettisivustojen suunnitteluun [35]. Termiä responsiivinen web-suunnittelu käytti ensimmäisenä Ethan Marcotte vuonna 2010 [32]. Lähestymistavan perimmäinen ajatus on peräisin nousevasta arkkitehtuurin koulukunnasta nimeltä Responsiivinen Arkkitehtuuri (Responsive Architecture), joka analysoi kuinka fyysiset tilat voivat vastata ohi kulkevien ihmisten läsnäoloon. Internetin kontekstissa tämä tarkoittaa kaikkien laitteiden käsittelyä eri näkökulmina samasta kokemuksesta. [39] Responsiivinen web-suunnittelu on kasvattanut suosiotaan viime vuosina merkittävästi ja vuonna 2013 sen arvioitiin olevan yksi suurimmista markkinoinnin trendeistä usean artikkelin ja suositun sivuston mukaan [7] [44] [29].

Responsiivisen web-suunnittelun avulla voidaan luoda web-sivusto, joka huomioi tulevat teknologiset kehitykset ja on helposti ylläpidettävä. Lisäksi responsiivisilla sivustoilla on yhtenäinen käyttäjäkokemus eri laitteiden välillä, jolloin se poistaa tarpeen erillisille laitespesifisille ratkaisuille. [39] Responsiivisen sivuston ylläpitoa helpottaa se, että sivustolla on vain yksi URL-osoite ja yksi HTML-koodi kaikille laitteille. Tällöin sisältöä päivitettäessä muutokset tarvitsee tehdä vain kerran (vrt. erilliset mobiilisivustot). Yhtenäinen käyttäjäkokemus saavutetaan muun muassa hyödyntämällä CSS3 mediakyselyitä (media queries) ulkoasun luomiseen siten, että sisältö mukautuu automaattisesti erikokoisille näytöille. Responsiivisen web-suunnittelun tavoitteena on saavuttaa sivuston hyvä luettavuus ja navigointi mahdollisimman pienellä määrällä sivuston lähentämistä (zoom) ja vierittämistä (scroll) eri suuntiin sen käytön aikana [35].

Responsiivinen web-suunnittelu sekoitetaan joskus Adaptiiviseen suunnitteluun (Adaptive Design), joka on samankaltainen, mutta erillinen lähestymistapa web-suunnittelulle. Adaptiivisen suunnittelun avulla voidaan myös luoda moderneja laitteen koon mukaan mukautuvia sivustoja. Molemmat näistä metodologioista hyödyntävät niin sanottuja pysäytyspisteitä (breakpoints), jotka toimivat raja-arvoina ulkoasun muuttamiselle ikkunan (viewport) koon muuttuessa. Adaptiivinen ja responsiivinen suunnittelu eroavat kuitenkin siinä kuinka sivu muuttuu raja-arvojen välillä.

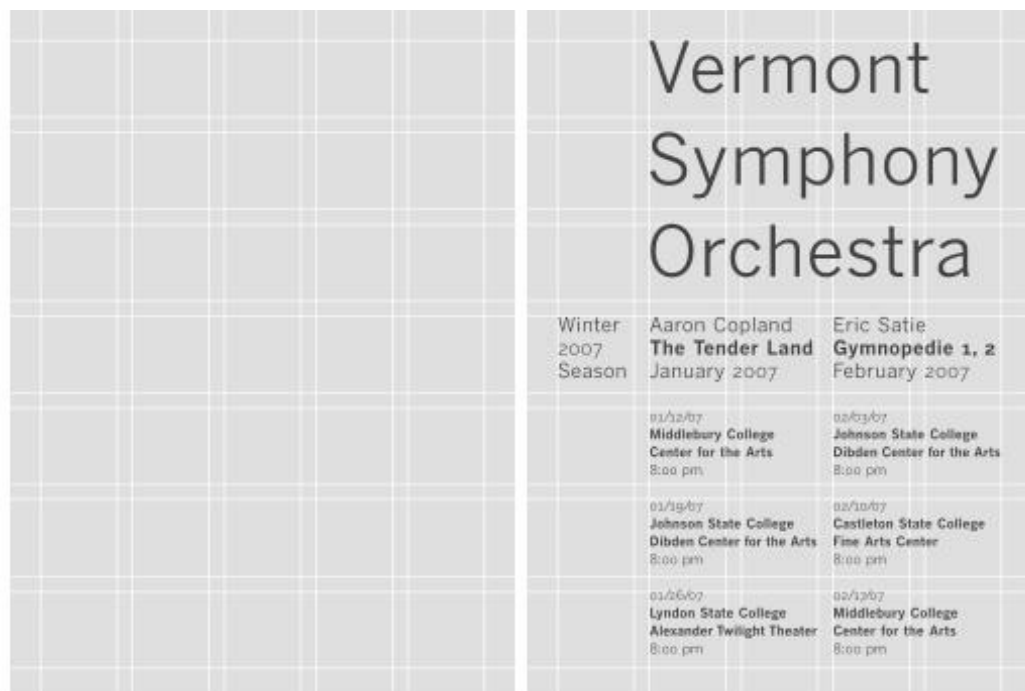
Adaptiivisessa suunnittelussa luodaan käytännössä erillisiä kiinteälevyisiä ulkoasuja, joita vaihdellaan raja-arvojen välillä, kun responsiivisessa suunnittelussa puolestaan luodaan yksi joustava ulkoasu, joka mukautuu ikkunan koon mukaan. [16] Responsiivisen suunnittelun avulla pystytään huomioimaan jatkuvasti vaihtuvat laitteiden näyttöjen koot paremmin erillisten kiinteälevyisten ulkoasujen (Adaptiivinen suunnittelu) ylläpitämiseen verrattuna.

2.2.1 Teknologiat ja toteutus

Responsiivinen web-suunnittelu yhdistää jo ennestään olemassa olevia teknologioita ja suunnittelumalleja kokonaisuudeksi, joka mahdollistaa käyttöliittymän mukautumisen käytössä olevan laitteen näytön kokoon. Teknisestä näkökulmasta responsiivinen web-suunnittelu koostuu kolmesta ydinosasta: joustavasta ruudukkopohjaisesta ulkoasusta (Flexible grid layout), joustavista kuvista (Flexible images) sekä mediakyselyistä (Media queries) [31, s.9].

Joustava ruudukkopohjainen ulkoasu

Typografinen ruudukko on vanha graafisessa suunnittelussa hyödynnetty konsepti. Sen tarkoituksena on luoda riveistä ja sarakkeista koostuva systeemi, jonka mukaan sisältö asetellaan näkyville. Ruudukkopohjia (Kuva 2) on alettu myöhemmin käyttämään yleisenä lähtöpisteenä myös web-suunnittelussa uuden ulkoasun luomiselle. [31, s. 14 - 15]



Kuva 2: Esimerkki ruudukkopohjaisesta ulkoasusta. Kuva julkaisusta [37].

Kiinteissä ruudukkopohjissa, joissa ulkoasulle on määritelty kiinteä pikselimääräinen leveys, syntyy kuitenkin hankaluuksia, kun sivua selataan erikokoisilla laitteilla. Tästä

syystä responsiivisessa web-suunnittelussa pyritään käyttämään joustavia ruudukkoita, joissa mitat annetaan kiinteiden yksiköiden sijaan suhteellisissa yksiköissä. Kun ulkoasun mitat on annettu suhteellisissa yksiköissä (*em*) pikseleiden (*px*) sijaan, ulkoasu muuttuu jos ikkunan leveyttä muutetaan.

Joustavat kuvat

Joustavat kuvat liikkuvat ja joustavat kun niiden ympäröivä joustava rakenne muuttaa kokoaan [35]. Yksinkertaisimmillaan tämä saavutetaan määrittelemällä kuvien *max-width* CSS ominaisuudelle arvo *100 %*. Tällöin vaikka kuva olisi leveämpi kuin sitä ympäröivä elementti, se skaalautuu automaattisesti korkeintaan ympäröivän elementin leveyseksi ja muuttaa kokoaan samassa suhteessa ympäröivän elementin koon muuttuessa. Tämä ominaisuus toimii hyvin kaikissa moderneissa nettiselaimissa. Vanhoissa selaimissa (Internet Explorer 7 ja vanhemmat) kuvat skaalautuvat kuitenkin huonosti ja niihin muodostuu artefakteja kuvan koon muuttuessa, mutta tähän ongelmaan on tarpeen tullen olemassa ratkaisuja [31, s. 50 - 51].

Mediakyselyt

Mediakyselyjen (Media query) avulla voidaan määrittellä, kuinka sivuston ulkoasu muuttuu ikkunan resoluution muuttuessa. Tätä on havainnollistettu kuvassa 3. Mediakyselyt on määritelty osana CSS3 standardia [57]. Mediakyselyt ovat ehdollisia lausekkeita, joiden avulla voidaan tunnistaa median tyyppi (esimerkiksi *screen*, *print*, *handheld*, jne.) sekä ikkunan ominaisuuksia, kuten leveys ja korkeus.



Kuva 3: Mediakyselyjen ideaa havainnollistava kuva. Kuva julkaisusta [58].

Mediakyselyillä voidaan määrittää pysäytyspisteitä (*breakpoint*) käyttäen *max-width* ja *min-width* ominaisuuksia, jolloin perinteisiä CSS ominaisuuksia voidaan määrittellä voimassa olevaksi tietylle resoluution vaihteluvälille. Yksinkertainen esimerkki mediakyselystä on esitetty kuvassa 4. Siinä määritellään, että kun mediatyyppinä on

screen ja ikkunan leveys on alle 1024 pikseliä, niin *body* elementin sisällä CSS ominaisuuden *font-size* arvo on 100 %.

```

1  @media screen and (max-width: 1024px) {
2      body {
3          font-size: 100%;
4      }
5  }
```

Kuva 4: Yksinkertainen esimerkki mediakyselystä.

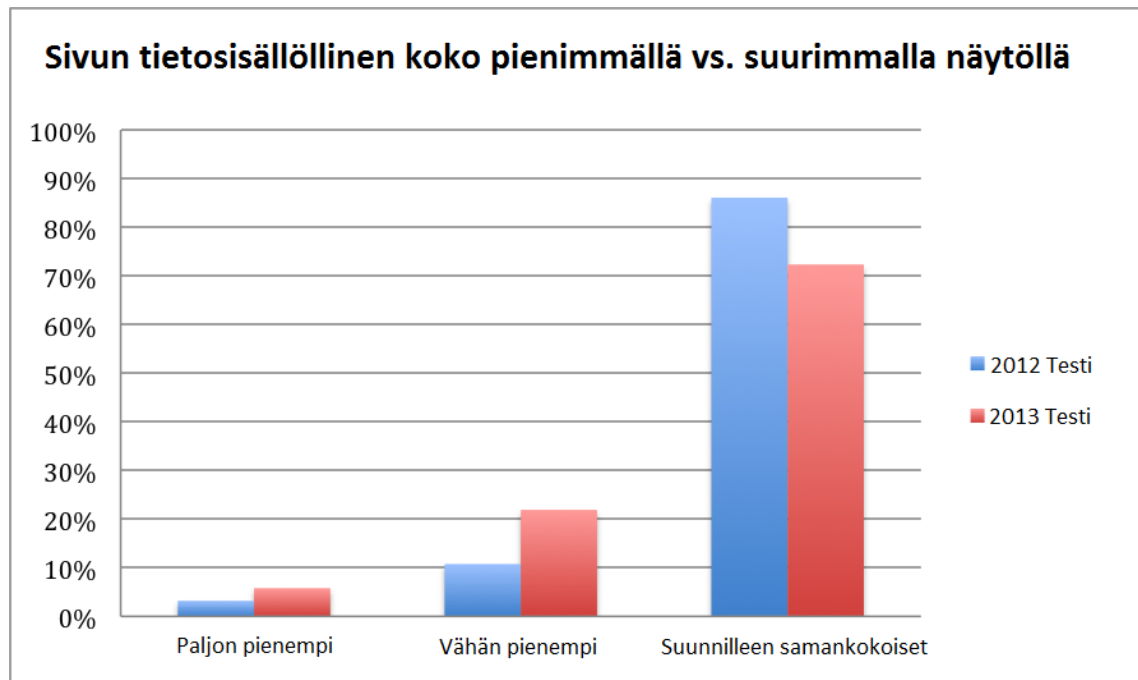
Responsiivisen sivuston toteuttamiselle on olemassa kaksi lähestymistapaa: sivusto voidaan tehdä joko mobiililähtöisesti (Mobile First) tai työpöytälähtöisesti (Desktop First). Molempien lähestymistapojen ideana on pohjimmiltaan se, että sivusto toteutetaan lähestymistavasta riippuen yhdelle resoluutiolle, minkä jälkeen huomioidaan muut resoluutiot valitsemalla sopivat pysäytyspisteet ja tavat sisällön näyttämiseksi mediakyselyiden avulla [35]. Jos sivuston arkkitehtuuri luodaan ensiksi ensisijaisesti mobiililaitteita varten (mobiililähtöisesti) ja tämän jälkeen lisätään sisältöä suurempia resoluutioita varten hyödyntäen mediakyselyitä, varmistetaan sivuston optimaalisesta suorituskyvystä mobiililaitteilla ja skaalautuvuudesta suurempiin ikkunoihin [12]. Työpöytälähtöisesti ulkoasu tehdään puolestaan ensiksi työpöytäresoluutioille, minkä jälkeen vastaavasti mietitään kuinka sisältö näytetään pienemmillä näytöillä [35].

2.2.2 Huomioita

Responsiiviseen web-suunnitteluun liittyy yhä jonkin verran ongelmia, koska etenkin vanhemmat nettiselaimet ja mobiililaitteet eivät tue kaikkia tarvittavia ominaisuuksia, kuten mediakyselyitä yhdenmukaisesti [35]. Useimpiin ongelmiin on kuitenkin olemassa JavaScript ohjelmointikielellä toteutettuja varasuunnitelma-tyyppisiä (fallback) ratkaisuja, joista käytetään termiä *polyfill* [24]. Ne ovat siis ratkaisuja, jotka simuloivat selaimien puutteellisia ominaisuuksia.

Responsiivisten sivustojen suorituskyyky on yksi keskustelua herättävä aihe. Sivujen hitaat latausajat ovat suurin suorituskyykyongelma responsiivisissa sivustoissa [35]. Esimerkiksi responsiiviset kuvat aiheuttavat ongelmia suorituskyykyyn, mikä on ongelma, johon ei ole vielä useista yrityksistä huolimatta löytynyt selkeää ratkaisua [26, s. 125]. Koska käyttöliittymä on yhtenäinen kaikille laitteille, mobiililaitteille ladataan tarpeettoman suuria kuvia, jotka latauksen jälkeen skaalataan sopivan kokoisiksi mobiilinäyttöä varten. Ongelma ei siis ole se miltä kuvat näyttävät vaan kuinka suuria ne ovat tietosisällöllisesti. Käyttöliittymän lataus kestää siis pienillä näytöillä tarpeettoman kauan eikä toistaiseksi ole helppoa tapaa ladata pienempiä kuvia mobiililaitetta käytettäessä.

Guy Podjarny on tutkinut responsiivisten sivustojen suorituskykyä suorittamalla kaksi testiä vuosina 2012 ja 2013. Testeissä hän latsi 347 responsiivista sivustoa eri näytön resoluutioilla ja mittasi ladatun tietosisällön kokoa ja sivuston latausnopeutta. Testeissä kävi ilmi (Kuva 5), että vaikka pienemmillä resoluutioilla näytetään paljon vähemmän tietoa suurempiin resoluutioihin verrattuna, ladattava tietosisältö on silti useimmiten lähes sama. [42] Tulos osoittaa, että responsiivisten sivustojen latausajoissa erityisesti mobiililaitteilla on paljon kehittämisen varaa, koska ladattava tietosisältö on useimmiten tarpeettoman suuri.



Kuva 5: Ladatun tietosisällön koko eri resoluutioilla, vuosina 2012 ja 2013. Kuva muokattu lähteestä [42].

Hyvän käyttäjäkokemuksen edellytyksenä responsiivisissa sivustoissa on, että suorituskykyyn panostetaan koko suunnittelu- ja kehitysprosessin ajan [35]. Uutta projektia aloittaessa on syytä huomioida, että perinteiseen web-suunnitteluun verrattuna responsiivista web-suunnittelua hyödyntävät projektit vievät noin 10 - 20 % enemmän aikaa [33]. Tätä kompensoi kuitenkin se, että erilliselle mobiilisivulle ei ole tarvetta ja sivuston ylläpitäminen on tehokkaampaa, koska muutokset tarvitsee tehdä vain yhteen koodipohjaan. Olemassa olevien järjestelmien muuttaminen responsiivisiksi vaatii myös usein paljon työtä, koska koko ulkoasun alla oleva pohja täytyy muuttaa joustavan ruudukkopohja-konseptin mukaiseksi [36].

2.3 Lähestymistapojen vertailu

Jokaisella lähestymistavalla mobiililaitteiden huomioimiselle on omat hyvät ja huonot puolensa, eikä mikään vaihtoehto ole kaikkiin tilanteisiin sopiva. Serrano et al. [48] vertailivat lähestymistapoja toisiinsa sovelluksiin liittyvien kriteerien perusteella.

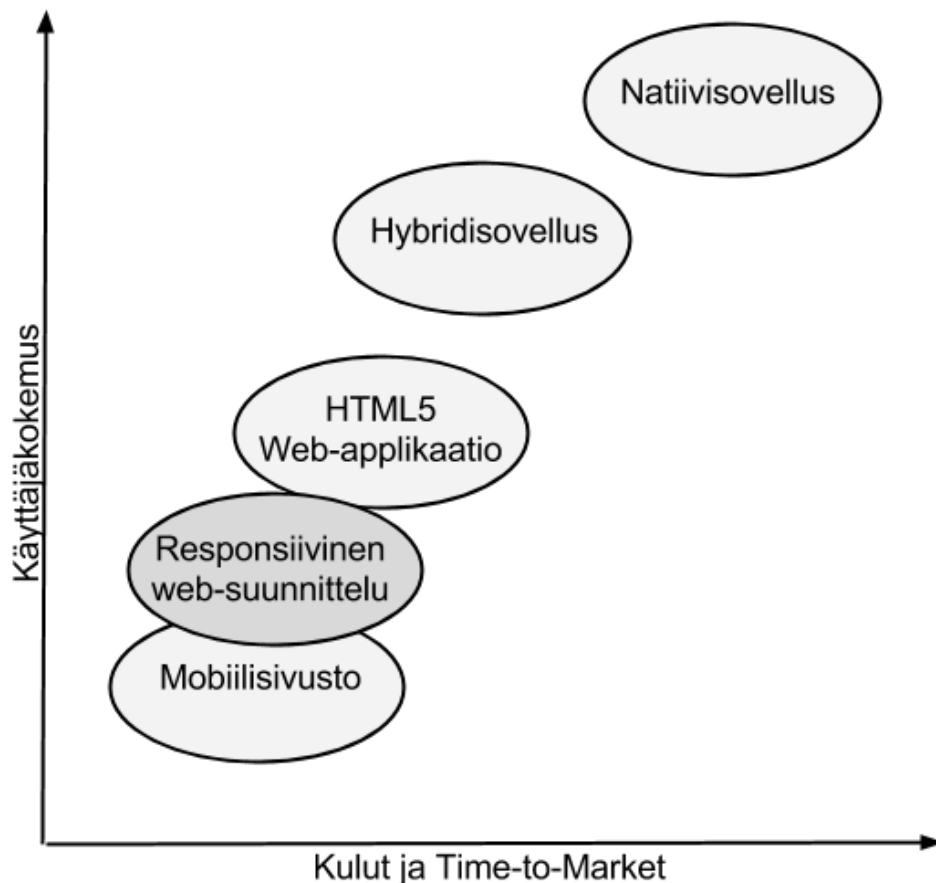
Vertailu on esitetty taulukossa 2. Responsiivista web-suunnittelua ei ole erikseen listattu, vaikka se onkin hyväksi todettu lähestymistapa mobiililaitteiden huomioimiselle. Tätä perusteltiin sillä, että responsiivisen web-suunnittelun hyvät ja huonot puolet muihin vaihtoehtoihin verrattuna ovat hyvin lähellä mobiileja web-sovelluksia [48].

Taulukko 2: Huomioitavat kriteerit ja eri lähestymistapojen vertailu [48].

Kriteeri	Natiivi	Hybridi	Web/Responsive
Työmäärä eri alustojen ja versioiden tukemiseksi	Suuri	Suuri	Pieni
Pääsy laiterajapintaan	Täysi	Täysi	Osittainen
Käyttäjäkokemus	Hyvä	Hyvä	Keskiverto
Suorituskyky	Erittäin hyvä	Erittäin hyvä	Hyvä
Päivitykset laitteessa	Tarvitaan	Tarvitaan	Ei tarvita
Julkaisun/levityksen helppous	Keskiverto	Keskiverto	Hyvä
Hyväksymissyklit	Pakollinen	Joissain tapauksissa	Ei tarvita
Sovelluskaupan hyödyntäminen rahastukseen	Käytettävissä	Käytettävissä	Ei käytettävissä

Mobiilistrategioiden vertailu on parhaillaan ajankohtainen ja aktiivinen puheenaihe. Vuonna 2012 tehdyn tutkimuksen [50] mukaan 94 % sovellusten kehittäjistä uskoi HTML5 teknologioiden olevan lopulta voittava lähestymistapa. Vuonna 2013 tehdyn tutkimuksen [51] mukaan HTML5 teknologia kasvattaa suosiotaan entisestään etenkin, kun on tarve palvella useita käyttöjärjestelmiä. Tutkimuksessa kuitenkin todettiin myös, että ei ole yhtä lähestymistapaa, joka palvelee kaikkia tilanteita, vaan paras lähestymistapa vaihtelee tilanteesta riippuen [51]. Kaikesta huolimatta natiivisovellukset ovat tähän asti vain kasvattaneet suosiotaan kuluttajien keskuudessa: vuonna 2014 tehdyssä tutkimuksessa [27] kävi ilmi, että 86 % mobiililaitteilla käytetystä ajasta käytetään natiiveissa mobiilisovelluksissa.

Lähestymistapoja on vertailtu useissa viime aikoina aiheesta kirjoitetuissa nettiartikkeleissa (esimerkiksi [17] [28] [43]) ja tieteellisissä julkaisuissa (esimerkiksi [48] [21]). Kuva 6 havainnollistaa yksinkertaisesti johtopäätöstä, johon vertailuissa yleisimmin päädytään: mikään lähestymistapa ei ole absoluuttisesti parempi muihin verrattuna, vaan oikea valinta vaihtelee projektikohtaisesti. Projektin budjetti, aikataulu sekä käyttäjäkokemukselle asetetut vaatimukset ovat usein suurimmat valintaan vaikuttavat kriteerit.



Kuva 6: Mobiilistrategioiden väliset kompromissit. Kuva muokattu lähteestä [17].

Paras käyttäjäkokemus saavutetaan natiivisovelluksien avulla, koska niiden avulla voidaan hyödyntää laiterajapintaa ja niiden suorituskyky on muihin lähestymistapoihin verrattuna parempi. Ne ovat kuitenkin alustariippuvaisia, joten niiden toteutus on merkittävästi kalliimpaa muihin lähestymistapoihin verrattuna. Natiivisovelluksien toteutus voi olla järkevää suurissa projekteissa joilla on suuri budjetti tai pienemmissä projekteissa joissa riittää, että sovellus tukee vain yhtä käyttöjärjestelmää ja joissa käyttäjäkokemus koetaan kriittisen tärkeäksi.

Hybridisovellukset ja mobiilit HTML5 web-sovellukset ovat alustariippumattomia ja kustannustehokkaita lähestymistapoja natiivisovelluksiin verrattuna. Hybridisovelluksien avulla päästään laiterajapintapääsyn ansiosta käyttäjäkokemuksessa lähelle natiivisovelluksia, mutta niiden suorituskyky ei vastaa natiivisovelluksien tasoa. Mobiilit HTML5 web-sovellukset ovat hyvä lähestymistapa, kun järjestelmä ei vaadi pääsyä laiterajapintaan.

Responsiivinen web-suunnittelu on mobiilisovelluksiin verrattuna mobiililaitteiden huomioimisen lähestymistapana lähimpänä mobiileja HTML5 web-sovelluksia, mutta se on alustariippumattomuuden lisäksi myös laiteriippumaton. Toisin sanoen se huomioi mobiililaitteiden lisäksi myös muut laitteet, kuten pöytäkoneet, tabletit ja kannettavat

tietokoneet. Tämä tekee responsiivisesta web-suunnittelusta erittäin kustannustehokkaan ja helposti ylläpidettävän vaihtoehdon mobiililaitteiden huomioimiseen erityisesti silloin, kun kyseessä olevaa järjestelmää käytetään myös muilla kuin mobiililaitteilla. Vaikka responsiivinen web-suunnittelu on vielä kehitysvaiheessa, se vastaa jo nyt hyvin nykyisiin suunnitteluongelmiin [3]. Sen heikkoutena muihin mobiililaitteiden huomioimisen lähestymistapoihin on kuitenkin suorituskyky.

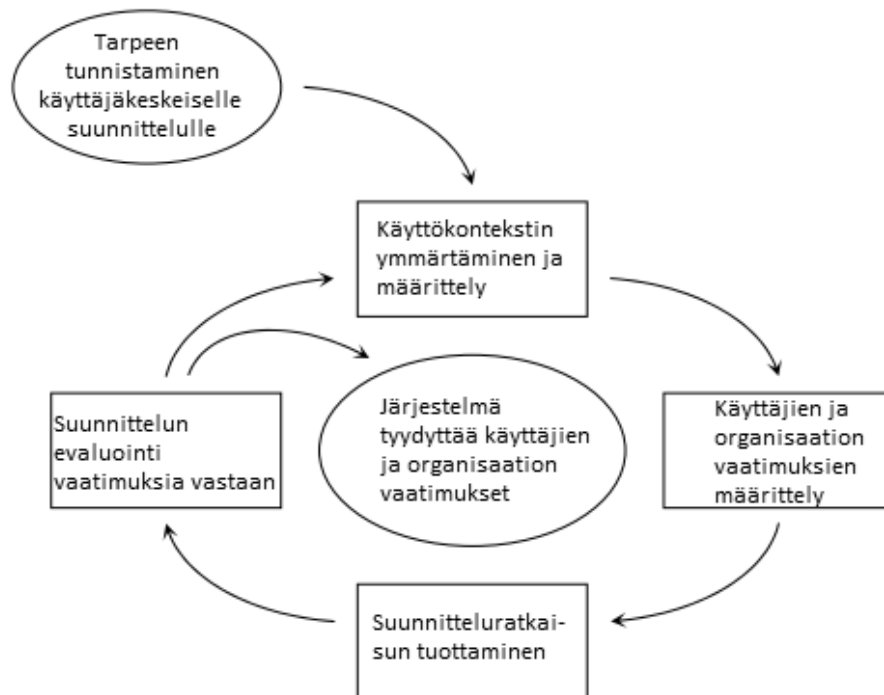
3. KÄYTTÄJÄKESKEINEN SUUNNITTELU JA PROSESSIT

Tässä työssä tehdään prototyyppi, joka toteutetaan hyödyntäen käyttäjäkeskeistä suunnittelua. Tässä luvussa käsitellään käyttäjäkeskeistä suunnittelua sekä siihen liittyviä prosesseja.

3.1 Käyttäjakeskeinen suunnittelu

Käyttäjakeskeinen suunnittelu (Ucer-centered Design, UCD) on lähestymistapa interaktiivisten järjestelmien kehittämiseen, joka keskittyy tekemään järjestelmistä hyviä käytettävyydeltään [23]. Se on joukko menetelmiä, jotka pyrkivät ottamaan käyttäjät mukaan järjestelmän suunnitteluprosessiin [1]. Käyttäjakeskeisen suunnittelun menetelmiä on useita, mutta kaikkia yhdistää käyttäjien tuominen keskeiseksi osaksi suunnitteluprosessia. Termiä käyttäjakeskeinen suunnittelu on käytetty ensimmäisiä kertoja jo vuonna 1986 [38] ja metodologia standardisoitiin vuonna 1999 ISO:n (the International Organization for Standardization) standardissa 13407 (*Human-centred design processes for interactive systems*) [23].

Käyttäjakeskeisessä suunnitteluprosessissa on perinteisesti kolme ydinvaihetta: tutkimus, suunnittelun toteutus ja evaluointi [59]. Tutkimusvaiheessa suunnittelijan tarkoituksena on muun muassa selvittää ketkä ovat tuotteen käyttäjät ja mitä heidän tarpeensa ovat. Suunnittelun toteutusvaiheessa tutkimuksessa tehtyjen löydösten pohjalta tehdään luonnos suunniteltavasta lopputuotteesta, joka voi projektista riippuen olla esimerkiksi käyttöliittymän prototyyppi tai informaatioarkkitehtuuri. Kun luonnos suunnitelmasta on valmis, se evaluoidaan hyödyntäen käyttäjiä ja toteutusta muokataan evaluoinnin pohjalta. [59] Tämä prosessi voidaan toistaa tarvittaessa useita kertoja, jotta päästään haluttuun lopputulokseen. Kuvassa 7 sama prosessi on esitetty hieman yksityiskohtaisemmin ISO:n standardin 13407 mukaisesti.



Kuva 7: Käyttäjakeskeisen suunnittelun prosessi. Kuva muokattu lähteestä [25].

Yksi käyttäjakeskeisen suunnittelun suuria hyötyjä on syvempi ymmärrys psykologisista, organisatorisista, sosiaalisista ja ergonomisista tekijöistä, jotka vaikuttavat järjestelmän käyttöön. Ottamalla käyttäjät mukaan suunnitteluprosessiin voidaan varmistua, että järjestelmä tulee vastaamaan sille asetettua tarkoitusta siinä ympäristössä, jossa sitä tullaan käyttämään. [1] Käyttäjakeskeisen suunnittelun vaikutuksista tehdyssä tutkimuksessa [56] noin 79 % vastanneista yrityksistä koki käyttäjakeskeisen suunnittelun parantavan tuotteen käytettävyyttä ja hyödyllisyyttä merkittävästi. Tutkimuksen mukaan yleisimmiksi käyttäjakeskeisen suunnittelun hyödyiksi mainittiin lisääntynyt asiakastyytyväisyys, tuotteen helppokäyttöisyyden parantuminen, vaikutukset myyntiin sekä vähentyneet asiakastukisoitot [56].

Käyttäjakeskeisen suunnittelun avulla suunnittelijat pystyvät lisäksi hallitsemaan käyttäjien odotuksia järjestelmästä paremmin. Kun käyttäjät ovat olleet mukana suunnittelemassa järjestelmää, he tietävät alusta asti mitä odottaa järjestelmältä ja tuntevat, että heidän ideansa ja ehdotuksensa on otettu huomioon suunnitteluprosessissa. Tästä seuraa usein korkeampi asiakastyytyväisyys ja sujuvampi järjestelmän käyttöönotto sille tarkoitetussa ympäristössä. [1]

3.2 Mobiilikäyttöliittymien suunnitteluperiaatteet

Käyttäjien ja suunnittelijoiden yhteistyö on tärkeää menestyksekkäiden käyttöliittymien luomisessa [20]. Perinteinen käyttöliittymäsuunnittelu ei kuitenkaan ole enää riittävää optimaalisen mobiilikäyttöliittymän suunnittelussa [9]. Vaikka monet käyttöliittymäsuunnittelun periaatteet pätevät yhtäläillä pöytäkoneilla ja mobiililaitteilla,

mobiilikäyttöliittymän suunnittelussa on myös paljon eroavaisuuksia [30]. Suuri osa näistä eroavaisuuksista johtuu mobiililaitteisiin liittyvistä rajoituksista, kuten pienestä ruudun koosta, erilaisista syötteen antamisen mahdollisuuksista, pienestä muistista ja lyhyestä akun kestosta [2]. Mobiililaitteiden pieni näyttö tekee tehokkaasta tiedon näyttämisestä ja navigoinnista halutun tiedon luokse vaikeaa, joten suunniteltaessa mobiililaitteiden käyttöliittymiä on erityisen tärkeää panostaa käyttöliittymien opittavuuden helppouteen, käytön tehokkuuteen ja nautinnolliseen käyttäjäkokemukseen [20].

Mobiililaitteilla tekstin luettavuus on merkittävästi heikompaa työpöytätietokoneisiin verrattuna pienestä näytön koosta johtuen. Tekstin luettavuutta näytöiltä tutkittiin jo ennen Internet-aikakautta tehdyssä tutkimuksessa [34], jossa havaittiin, että lukunopeus on noin 30 % nopeampaa paperilta kuin näytöltä. Teknologian kehityksen myötä suurempien näyttöjen luettavuus on kasvanut merkittävästi, mutta pienillä näytöillä luettavuus on edelleen kuitenkin suuri ongelma. Tästä syystä tekstin määrä kannattaa pyrkiä minimoimaan mobiilikäyttöliittymissä ja näyttää oleellinen tieto aina ensiksi [19]. Tekstiä kannattaa pyrkiä myös korvaamaan sopivilla ikoneilla sopivissa paikoissa, mikä vähentää luettavan tekstin määrää [ibid.].

Mobiililaitteilla syötteen antaminen on hankalaa tai ainakin hyvin erilaista pöytäkoneisiin verrattuna, koska mobiililaitteiden näppäimistöt ovat usein hyvin rajallisia ja näppäimet ovat pieniä. Tämä tekee tekstin kirjoittamisesta ja erityisesti pitkien lomakkeiden täyttämisestä vaikeaa. [15] Tekstinsyöttökenttien sijaan kannattaa pyrkiä hyödyntämään esimerkiksi erilaisia monivalintavalikoita [19].

Käyttäjien aiheuttamien virheiden estäminen ja niistä palautuminen on erityisen tärkeää mobiililaitteilla [19]. Mobiililaitteissa on vaihtelevia ratkaisuja yleisille tarpeille, kuten esimerkiksi ”edellinen”-näppäimelle. Osassa laitteista ”edellinen”-näppäintä ei ole ollenkaan ja niissä tapauksissa kun kyseinen näppäin löytyy, kaikki käyttäjät eivät osaa käyttää sitä oikein. Tällöin esimerkiksi virheistä palautuminen voi olla hankalaa. [15] Virhetapauksissa on tärkeää tarjota helppo edellisen toiminnon palauttava toiminnallisuus sekä ymmärrettävä ja selkeä virheilmoitus [19].

Yksi huomioitava asia on myös, että mobiililaitteiden tapauksessa käyttäjä saattaa kiinnittää vain rajallisen määrän huomiota käyttöliittymään, esimerkiksi liikkeellä ollessaan tai sosiaalisessa tilanteessa [9]. Tutkimusten mukaan laitteita käytetään lisäksi yhä enemmän yhtä aikaa toistensa kanssa. Googlen 2012 tekemässä tutkimuksessa [52] 81 % osallistujista käytti älypuhelimiaan samalla kun katsoivat televisiota ja 66 % piti älypuhelimia saatavilla käyttäessään työpöytätietokonetta. Käyttöliittymän käyttö voi siis usein olla mobiililaitteilla toissijainen tehtävä, jolloin käyttöliittymän yksinkertaisuus ja käytettävyys nousevat suunnittelun keskiöön.

Eräs työpöytätietokoneille ja mobiililaitteille suunnittelun suurimpia eroja on käyttökontekstin aiheuttamat vaikutukset [30]. Mobiililaitteet ovat erittäin personoituja laitteita, jotka ovat aina päällä, aina mukana ja suurimman osan ajasta yhteydessä Internetiin. Lisäksi niissä on usein hyödyllisiä sensoreita, jotka voivat havaita sijainnin, liikettä, kiihtyvyyttä, asentoa, läheisyyttä, ympäristöllisiä olosuhteita ja muita ominaisuuksia. Tämä tekee mobiililaitteiden käytöstä monessa suhteessa tehokkaampaa pöytäkoneisiin verrattuna, vaikka teknisen suorituskyvyn perusteella työpöytätietokoneet ovat yhä tehokkaampia. [16] Mobiililaitteet muuttavat Internetin käyttökontekstia, mikä luo käyttäjille uusia tarpeita, joka puolestaan asettaa haasteita sovelluksien kehittäjille.

Mobiilikäyttöliittymää suunniteltaessa on myös hyvä huomioida, että mobiilit Internet yhteydet ovat usein hitaampia kiinteisiin kaapeliverkkoihin verrattuna. Tämä hidastaa tiedon esillesaantia, mikä voi vaikuttaa negatiivisesti käytettävyyteen. Tiedon esillesaannin nopeus vaikuttaa käyttäjäkokemukseen mobiililaitteita käytettäessä erityisen paljon, koska mobiililaitteiden käyttäjillä on usein suuremmat ja tavoitteellisemmat aikomukset pöytäkoneiden käyttäjiin verrattuna. Yleinen tavoite on esimerkiksi jonkun käyttäjän kontekstiin liittyvän yksittäisen tiedon selvittäminen. [15]

3.3 Tutkimusmenetelmät

Erilaisia käyttäjäkeskeisen suunnittelun prosessissa käytettäviä menetelmiä ovat esimerkiksi ryhmähaastattelut, yksilöhaastattelut, kyselyt, paikan päällä tapahtuvat havainnoinnit ja käytettävyydestaukset. Niitä käytetään eri vaiheissa käyttäjäkeskeisen suunnittelun prosessia. Esimerkiksi haastattelujen ja havainnoinnin avulla voidaan kerätä tietoa käyttökontekstista ja suunnittelun vaatimuksista aikaisessa vaiheessa suunnittelusykliä. Käytettävyydestien ja kyselyiden avulla voidaan puolestaan suunnittelusyklin lopuksi testata suunnittelun toteutunutta käytettävyyttä. [1]

3.3.1 Contextual Inquiry

Tärkeä vaatimus käyttäjäkeskeiselle suunnittelulle on kerätä tietoa käyttäjien tarpeista ja käyttökontekstista suunnittelun alkuvaiheissa [4]. Perinteisien haastattelumenetelmien avulla on usein mahdotonta saada selville kaikkia yksityiskohtia työskentely-ympäristöihin ja -tapoihin liittyen. Työskentelyyn liittyy aina paljon ulospäin vähemmän ilmeisiä yksityiskohtia, jotka ovat kuitenkin tekijälle itselleen niin itsestään selviä, että niiden mainitseminen ei välttämättä käy edes mielessä. Yksi hyvä tapa saada selville näitä yksityiskohtia on mennä käyttäjän luokse työskentely-ympäristöön, havainnoida työskentelyä ja keskustella työskentelystä käyttäjien kanssa. Tämä on Contextual Inquiry -menetelmän perusajatuksena, mikä on yksi osa isompaa tutkimusmenetelmäkokonaisuutta nimeltä Contextual Design [5].

Contextual Inquiry -menetelmä tarjoaa erilaisia käyttäytymismalleja, joiden tarkoitus on vähentää erilaisten sääntöjen ulkoa opettelua ja tukea luonnollista ihmisten vuorovaikutusta tutkimustilanteessa. Yksi yleisesti käytetty malli on mestari-oppipoika -malli, jonka tarkoitus on rakentaa vahva perusta työympäristöstä oppimiselle. Mallissa suunnittelija haluaa oppia käyttäjän työstä, aivan kuten oppipoika haluaa oppia taidon mestarilta. Mestari-oppipoika -mallin avulla käyttäjä tulee helpommin tietoiseksi siitä, mitä he tekevät ja miksi he tekevät niin, mikä voi paljastaa suunnittelun kannalta oleellisia yksityiskohtia käyttäjän työstä. Oppipoika-mestari asetelma on luonnollinen malli tiedon välittämiseen tekemisestään myös henkilöille, jotka eivät muuten ole erityisen opettamiseen orientoituneita. [5]

Contextual Inquiry -menetelmässä on neljä periaatetta, joiden avulla tietoa voidaan kerätä tilanteelle optimaalisella tavalla. Nämä ovat konteksti, kumppanuus, tulkinta sekä huomion keskittäminen. Kontekstilla tarkoitetaan työympäristöön menemistä ja työskentelyn toteutumisen näkemistä, mikä on olennaisin vaatimus Contextual Inquiry -menetelmässä. Kumppanuuden tarkoituksena on saada haastattelija sekä itse käyttäjä ymmärtämään työskentelytavat ja -ympäristö. Tulkinnalla tarkoitetaan sitä, että havainnoinnin avulla kerätyt faktat eli havainnot käyttäjän työskentelystä ovat vain lähtökohta kehitykselle ja todellinen kehitys muodostuu havaintojen tulkinnan pohjalta. Tulkintojen jakaminen on tärkeää, jotta päättelyketjun pätevyys voidaan varmistaa. Huomion keskittämällä tarkoitetaan sitä, että suunnittelijan tulee johdattaa käyttäjä keskustelemaan asioista, jotka ovat olennaisia suunnittelun kannalta. [5]

Periaatteet ohjaavat tiedonkeräystä suuntaan, joka palvelee parhaiten ratkaistavaa ongelmaa. Tarkka tiedonkeräysprosessi määräytyy kuitenkin tapauskohtaisesti suunnitteluongelman ja työn luonteen mukaan. Haastattelutilanteelle on kuitenkin olemassa rakenne, joka palvelee useimpia Contextual Inquiry -menetelmällä tehtyjä tutkimuksia. Haastattelu alkaa yleensä niin sanotulla tavanomaisella haastattelulla, jossa tarkoituksena on, että haastattelija ja haastateltava saavat hyvän kontaktin toisiinsa. Tähän vaiheeseen kuuluvat esimerkiksi esittäytymiset ja luottamuksellisuudesta kertominen, lupien kysyminen ja tutkimuksesta kertominen. Tämän jälkeen tulee yleensä lyhyt niin sanottu siirtymävaihe, jossa kerrotaan haastattelun käytännöt. Siirtymävaihetta seuraa itse kontekstuaalinen haastattelu, joka on haastattelun pääosa. Siinä asiakas aloittaa työskentelyn ja haastattelija toimii oppipoikana, havainnoi, kysyy kysymyksiä ja tekee tulkintaehdotuksia. Viimeinen vaihe on yhteenveto, jossa käydään lyhyesti läpi haastattelussa ilmenneet asiat. [5]

Hyvä Contextual Inquiry -menetelmällä tehty haastattelu ei niinkään vaadi tiettyjen sääntöjen noudattamista, vaan enemmän oikean tyyppisen roolin omaksumista haastattelun aikana. Oppipoika-malli on hyvä lähtökohta haastattelun toteuttamiselle. Lisäksi menetelmän periaatteet muokkaavat käyttäytymistä suuntaan, joka mahdollistaa entistä paremman suunnittelua tukevan tiedon keräämisen. [5]

3.3.2 Prototypointi

Käyttäjakeskeisen suunnittelun yksi perimmäisistä ajatuksista on, että käyttäjät määrittelevät lopulta mikä suunnitteluratkaisu toimii ja mikä ei. Jotta suunnitteluratkaisua on järkevää lähteä toteuttamaan, sen tulisi joko yksinkertaistaa käyttäjän työtä, vähentää työhön liittyviä vaikeuksia tai muulla tavalla tehostaa työkäytäntöjä. Prototypointi on työkalu, jonka avulla käyttäjälle voidaan kommunikoida suunnitteluratkaisu ja testata täyttääkö suunnitteluratkaisu sen tarkoituksen käyttäjän näkökulmasta. [5]

Prototyypin tarkoituksena on varmistaa, että kerätyn tiedon pohjalta tehdyt tulkinnat vastaavat käyttäjien tarpeisiin ennen kuin käytetään paljon aikaa järjestelmän suunnitteluun ja kehitykseen. Prototyypin avulla voidaan testata ideoiden soveltuvuutta ja hyödyllisyyttä käyttäjien kanssa, mahdollistaa käyttäjien osallistaminen suunnitteluprosessiin sekä käyttäjien omien ideoiden testaamisen. [49] Prototyypin avulla käyttäjät näkevät selkeästi minkälaisia seurauksia erilaisilla suunnitteluratkaisuilla on, jolloin he pystyvät reagoimaan niihin asianmukaisesti [5].

Prototyypit jaotellaan kahteen eri kategoriaan: matalan tason prototyypeihin, jotka perustuvat useimmiten paperiprototyypeihin, ja korkean tason prototyypeihin, jotka ovat ohjelmoituja tai ohjelmiston avulla luotuja. Käyttöliittymän suunnittelun elinkaaren aikana on yleensä syytä hyödyntää molempien kategorioiden prototyypejä. [49]

Matalan tason prototyypejä voidaan piirtää käsin, mutta lisäksi niitä voidaan esimerkiksi piirtää jonkin kuvankäsittelyohjelman avulla ja tulostaa käyttäjätestausta varten paperille. Matalan tason prototyypejä voidaan käyttää ideoiden, näyttöjen ulkoasun ja erilaisten suunnitteluvaihtoehtojen esittämiseen. Niiden avulla voidaan antaa karkeaa kuvaa käyttöliittymän ulkoasusta ja tuntumasta käyttäjille, mutta tehtävien suorituksesta ja käyttöliittymän toiminnallisuudesta voidaan tarjota kuva vain hyvin rajatulla tasolla. [49]

Contextual Inquiry -menetelmässä matalan tason prototyypejä voidaan hyödyntää kontekstuaalisen haastattelun yhteydessä yhdessä suunnitteluun asiakkaan kanssa. Haastattelijan tehtävänä on tässä tilanteessa tarjota vuorovaikutuksen perusteella haastateltavalle teknologian näkökulmasta mahdollisia tutkittavia vaihtoehtoja. Käyttäjä puolestaan auttaa haastattelijaa suunnittelussa pohtimalla haastattelijan tarjoamia vaihtoehtoja, yhdistää ne kokemukseensa työn tekemisestä ja kertoo mikä vaihtoehto voisi toimia ja mikä ei. [5]

Korkean tason prototyypejä tarvitaan tarkemman toiminnallisuuden demonstroimiseen. Ne ovat ohjelmoituja tai jonkin ohjelmiston avulla luotuja ja tarjoavat toiminnallisen version järjestelmästä, jonka kanssa käyttäjä voi olla vuorovaikutuksessa. Korkean tason prototyypin tulee tuntua ja käyttäytyä pääpiirteissään aivan kuten lopputuotteen ja

sitä tulee voida käyttää lopputuotteen markkinoinnin tukena. Koska korkean tason prototyypeillä voidaan demonstroida tarkempaa toiminnallisuutta, niiden avulla voidaan testata vaatimuksia, joiden testaus ei ollut mahdollista matalan tason prototyypeillä sekä havaita puuttuvia toimintoja. Korkean tason prototyyppien toteutus on kuitenkin enemmän aikaa vievää matalan tason prototyyppeihin verrattuna eikä niitä pystytä muokkaamaan testauksen aikana yhtä helposti. [49]

4. PROTOTYYPIN SUUNNITTELUPROSESSI

Tässä luvussa käsitellään järjestelmää, johon opinnäytetyö perustuu sekä käydään läpi prototyypin suunnitteluun vaikuttavat tekijät. Lisäksi luvussa tehdään katsaus toteutusteknisiin ratkaisuihin, joiden pohjalta prototyyppi toteutettiin.

4.1 CRM-järjestelmät

Asiakkuudenhallinta eli CRM (Customer Relationship Management) termi on saanut alkunsa IT-ammattilaisten yhteisössä 1990-luvun puolivälissä. Termille on olemassa useita määritelmiä, koska sen merkitys vaihtelee jonkin verran eri yritysten ja toimialojen välillä. Toisille CRM tarkoittaa tietovarastointia tai tiedon louhintaa, toisille se on Help Desk -alusta ja jotkut pitävät sitä yhtenä verkkokaupparatkaisuna. Yleisellä tasolla CRM-järjestelmillä tarkoitetaan kuitenkin usein teknologiapohjaisia asiakasratkaisuja. [40] Yksi määritelmä CRM termille on ”informaation mahdollistama asiakassuhdemarkkinointi” [41].

Asiakkuudenhallinta on tärkeää, koska se mahdollistaa tehokkaamman tiedon hyödyntämisen asiakkaiden ymmärtämisessä ja asiakassuhdemarkkinoinnin tehostamisessa. CRM-järjestelmät toimivat strategisena siltana tietotekniikan ja pitkäaikaisiin asiakassuhteisiin ja kannattavuuteen tähtäävien markkinointistrategioiden välillä. [41]

CRM-järjestelmien kysyntä on ollut erittäin suurta viime aikoina ja niiden markkinoiden kasvu jatkuu yhä. Vuonna 2013 CRM-järjestelmien markkinat kasvoivat 13,7 %. Nykyisin on tarjolla myös SaaS (Software-as-a-Service) -mallilla toimivia CRM-järjestelmiä, joiden tuottama liikevaihto kattoi 41 % markkinoiden kokonaisliikevaihdosta vuonna 2013. Yhtenä SaaS-mallin etuna pidetään niiden helppoa käyttöönottoa. Suurimpia CRM-toimittajia vuonna 2013 olivat salesforce.com (16,1 % markkinaosuus), SAP (12,8 % markkinaosuus) ja Oracle (10,2 % markkinaosuus). [14]

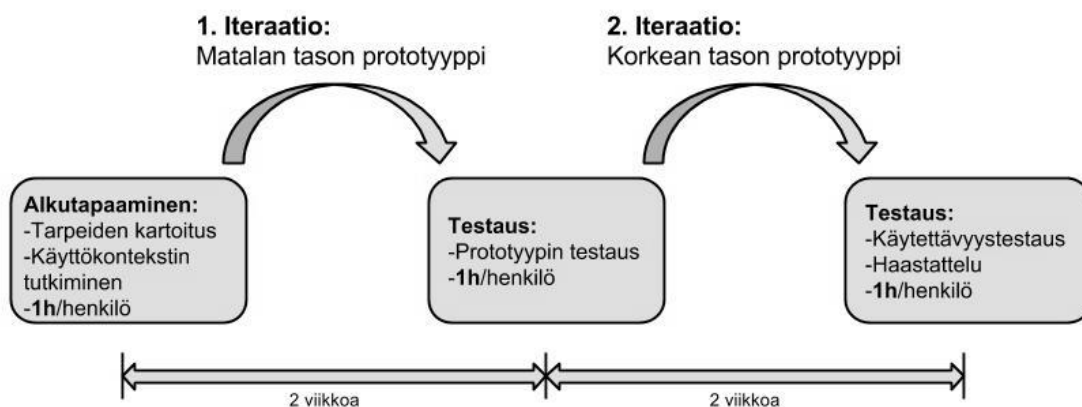
4.2 Tapaus: web-pohjainen CRM-järjestelmä

Työssä kehitettävä CRM-järjestelmä on erillinen täysin asiakkaan tarpeiden mukaan räätälöity web-pohjainen järjestelmä. Asiakas käyttää järjestelmää laajasti moniin eri liiketoimintaa tukeviin tarkoituksiin. Järjestelmässä on mahdollista hallita esimerkiksi tuotteita, kontakteja, tarjouksia ja kauppvoja näiden omissa osioissaan. Järjestelmän käyttäjäryhmiä ovat myyjät, myyntipäälliköt, aluepäälliköt. Järjestelmässä on käyttäjäryhmiin perustuvat pääsyräjoitukset järjestelmän eri osioihin.

4.3 Prototyypin toteutus

Ensimmäisessä luvussa tehdyn vertailun pohjalta tässä työssä tehtävän prototyypin lähestymistavaksi valittiin responsiivinen web-suunnittelu. Responsiivisen web-suunnittelun avulla voidaan luoda käyttöliittymä siten, että se mukautuu käytössä olevan laitteen näytön kokoon. Lähestymistavan valinta rajautui mobiiliin HTML5 web-sovellukseen ja responsiiviseen web-suunnitteluun sillä perusteella, että opinnäytetyöhön käytettävä aika ei mahdollista hybridi- tai natiivisovelluksen tekemistä, eikä järjestelmässä ole tarvetta laiterajapinnan hyödyntämiseen mobiilisovelluksessa. Vaikka mobiilin HTML5 web-sovelluksen avulla pystytään optimoimaan paremmin käyttäjäkokemusta responsiiviseen web-suunnitteluun verrattuna, sen toteuttaminen olisi hankaloittanut ylläpitoa tulevaisuudessa. Responsiivisen web-suunnittelun avulla olemassa olevaa koodipohjaa voitiin suhteellisen helposti muokata siten, että käyttöliittymästä tehtiin responsiivinen, jolloin se huomioi mobiililaitteet aiempaa paremmin.

Asiakkaan käytössä olevasta CRM-järjestelmästä toteutettiin mobiililaitteita tukeva responsiivinen käyttöliittymäprototyyppi. Prototyyppi toteutettiin muutamasta järjestelmän eniten käytetystä näkymästä käyttäjäkeskeisiä suunnitteluperiaatteita hyödyntäen. Käyttäjätutkimus toteutettiin kolmessa vaiheessa, jotka järjestettiin noin kahden viikon välein toisistaan kuvassa 8 esitetyn rakenteen mukaisesti.



Kuva 8: Tutkimuksen rakenne.

Tutkimuksen ensimmäisessä vaiheessa tutkittiin järjestelmän käyttöön liittyviä tarpeita ja selvitettiin eniten käytetyt toiminnot erityisesti mobiilikäytön näkökulmasta. Kerätyn tiedon pohjalta tutkimuksessa luotiin prototyyppi eniten käytetyistä näkymistä siten, että käyttäjien tarpeet ja mobiililaitteiden asettamat rajoitteet oli pyritty huomioimaan. Prototyypistä luotiin ensiksi matalan tason versio, jota testattiin toisessa vaiheessa ja testauksen avulla kerätyn tiedon pohjalta luotiin iteroinnin jälkeen vielä korkean tason

prototyyppi. Korkean tason prototyypin käyttöä mobiililaitteilla testattiin lopuksi rinnakkain alkuperäisen käyttöliittymän kanssa ja tuloksia verrattiin keskenään.

Korkean tason prototyypin käyttöliittymä tehtiin mobiililähtöisesti (Mobile First), eli käyttöliittymän ulkoasu ja sisältö optimoitiin ensiksi älypuhelimia varten. Muilla resoluutioilla sisältö näytettiin prototyypissä käytössä olevan järjestelmän mukaisesti. Seuraava askel prototyypin kehittämisessä olisi ollut testata järjestelmää useamman kokoisilla laitteilla ja alkaa miettiä miten sisältöä näytettäisiin kun leveyttä kasvatetaan, mutta kyseinen prosessi rajattiin tämän opinnäytetyön ulkopuolelle.

Responsiivisen käyttöliittymän toteutukseen valittiin pohjaksi Twitter Bootstrap -ohjelmistokehys, koska se oli tätä työtä tehdessä kehittynein ja eniten käytetty vaihtoehto tähän tarkoitukseen. Se tarjoaa valmiin joustavan mediakyselyitä hyödyntävän ruudukkopohjaisen rakenteen ulkoasulle yleisesti hyväksi todetuilla pysäytyspisteillä, pohjatyylittelyt useimmille HTML elementeille sekä lukuisia apuluokkia, jotka auttavat näyttävän ulkoasun luomisessa nopeasti. Kaikki ohjelmistokehysten tarjoamat toiminnot on mahdollista ylikirjoittaa ja kustomoida omiin tarkoituksiin, mikä tekee siitä erittäin joustavan työkalun responsiivisen käyttöliittymän korkean tason prototypointiin. Ohjelmistokehyksestä otettiin käyttöön uusin versio, joka tätä työtä tehdessä oli 3.2.0 [55].

5. VAIHE 1 – TARPEIDEN KARTOITUS

Tutkimuksen ensimmäisessä vaiheessa tutkittiin käytössä olevan CRM-järjestelmän käyttöön liittyviä tarpeita ja selvitettiin järjestelmän eniten käytetyt toiminnot erityisesti mobiililaitteilla käytettäessä. Lisäksi ensimmäisessä vaiheessa luonnosteltiin mobiilikäyttöliittymää ja toteutettiin matalan tason prototyyppi käyttöliittymästä hyödyntäen paperiprototyyppejä. Ensimmäisen vaiheen tutkimuskysymyksenä oli ”Mitkä järjestelmän toiminnot koetaan hyödyllisimmiksi mobiilikäytössä ja mitä tarpeita näihin liittyy?”.

5.1 Menetelmä

Tutkimuksen ensimmäisessä vaiheessa käytettiin Contextual Inquiry - tutkimusmenetelmää. Se on tutkimusmenetelmä tehtävien ja käytön tarpeiden tutkimiseen erityisesti olemassa olevan järjestelmän jatkokehityksessä, joten se soveltuu kyseessä olevaan tapaukseen hyvin. Menetelmässä tarkkaillaan käyttäjien työn suorittamista ja tehdään havaintoja työhön ja toimintaympäristöön liittyen. Tässä tutkimuksessa hyödynnettiin Contextual Inquiry -menetelmän ns. oppipoika-mestari -mallia, koska se antaa hyvän viitekehyksen työympäristöstä oppimiselle. Tutkimuksen rakenne mukailee Contextual Inquiry -menetelmän mukaista rakennetta. Contextual Inquiry -menetelmää ja sen rakennetta on kuvattu tarkemmin luvussa 3. Tässä tutkimuksessa keskustelua ohjattiin aktiivisesti siihen suuntaan, että mobiilikäyttöön liittyvät tarpeet nousivat esille.

Contextual Inquiry -osuuden jälkeen tutkimuksen ensimmäisessä vaiheessa ideoitiin mobiilikäyttöliittymää ja tehtiin luonnoksia hyödyntäen paperiprototyyppointia. Koska järjestelmän eri toimintojen käyttöliittymät ovat hyvin samankaltaisia, luonnostelu tehtiin yleisellä tasolla siten että se toimii pohjana usealle järjestelmän näkymälle. Luonnosten avulla tehtävän ideoinnin sekä Contextual Inquiry -menetelmän avulla kerätyn tiedon pohjalta tutkimuksen ensimmäisen vaiheen lopputuotteena tehtiin matalan tason prototyyppi tulevasta mobiilikäyttöliittymästä.

Tutkimusta varten rekrytoitiin 8 osallistujaa, jotka on listattu taulukossa 3. Järjestelmän suurin käyttäjäryhmä, jota varten mobiilikäyttöliittymä ensisijaisesti tehdään, on Myyjät ja muita ryhmiä ovat Myyntipäälliköt ja Aluepäälliköt. Myyjiä rekrytoitiin eniten, koska suurimpana käyttäjäryhmänä heidän tarpeet olivat ensisijaisessa asemassa prototyypin suunnittelussa. Muitakin käyttäjäryhmiä rekrytoitiin tutkimukseen, koska järjestelmän kehittämisen kannalta oli hyödyllistä ymmärtää, miten heidän työnsä eroaa myyjien työstä ja miten se vaikuttaa järjestelmän käyttöön.

Taulukko 3: Ensimmäisen vaiheen osallistujat.

#	Käyttäjryhmä
1	Myyjä
2	Myyjä
3	Myyntipääällikkö
4	Myyjä
5	Aluepääällikkö
6	Myyjä
7	Myyjä
8	Myyntipääällikkö

Tutkimus toteutettiin myyjien toimipisteillä, joissa tutkimus järjestettiin osallistujille sopivana ajankohtana. Tutkimus nauhoitettiin kannettavan tietokoneen mikrofonilla ja tätä varten osallistujilta kysyttiin lupa erillisellä lomakkeella ennen tutkimuksen aloittamista. Tutkimukselle varattiin aikaa yhteensä 60 minuuttia osallistujaa kohden. Ajan suunniteltiin jakautuvan siten, että Contextual Inquiry -menetelmän mukainen vaihe kestäisi noin 40 minuuttia ja loput 20 minuuttia varataan mobiilikäyttöliittymän ideointiin ja luonnoksien tekemiseen. Luonnostelu toteutettiin haastattelijan toimesta yhteistyössä haastateltavan kanssa ja sen tarkoitus oli konkretisoida keskusteltavia suunnitteluideoita.

Tutkimuksen tuloksia analysoitiin yksinkertaistetulla versiolla aineistoläheisestä laadullisesta eli induktiivisesta sisällönanalyysistä. Tämän menetelmän tarkoitus on järjestää aineisto tiiviiseen ja selkeään muotoon kadottamatta sen sisältämää informaatiota. Aineiston laadullinen käsittely perustuu loogiseen päättelyyn ja tulkintaan, jossa aineisto aluksi hajotetaan osiin, käsitteellistetään ja kootaan uudestaan uudella tavalla loogiseksi kokonaisuudeksi. Analyysin pohjalta pyritään lopulta saamaan tutkittavasta ilmiöstä kuvaus tiivistetyssä ja yleisessä muodossa. [54]

Analyysissä Contextual Inquiry -menetelmällä kerätty tieto jaoteltiin ensiksi sisällönanalyysin rakenteen mukaisesti pelkistetyiksi ilmaisuiksi. Tässä tapauksessa pelkistetty ilmaus tarkoittaa mobiilikäyttöä ajatellen hyödylliseksi koettua järjestelmän toimintoa. Esimerkki pelkistetyn ilmauksen luomisesta on esitetty taulukossa 4.

Taulukko 4: Esimerkki pelkistetyn ilmauksen luomisesta.

Alkuperäinen ilmaus	Pelkistetty ilmaus
"Mun täytyy pystyä luomaan kontakti nopeasti, jos saan jos joku antaa mulle yhteystiedot."	Kontaktin luominen

Seuraavaksi pelkistetyt ilmaukset listattiin ja jaoteltiin sopiviin alaluokkiin etsimällä niistä samankaltaisuuksia. Tässä tapauksessa alaluokittelu tehtiin sen perusteella, missä järjestelmän näkymässä kyseinen toiminto tehdään.

Alaluokittelun lisäksi ilmaukset jaoteltiin myös toiminnon tarkoituksen mukaan eli onko kyseessä tietoa muokkaava (M) vai jonkin tiedon tarkistamista (T) koskeva toiminto. Tämä tieto ilmaisee hyvin, minkä tyyppisiä toimenpiteitä mobiililaitteilla koetaan tarpeelliseksi tehdä.

Lopuksi tutkimuksessa tehtiin matalan tason prototyyppi hyödyllisimmiksi koetuista näkymistä. Matalan tason prototypointia, syitä sen käytölle ja siitä saatavia hyötyjä on kuvattu luvussa 3. Tässä tutkimuksessa hyödynnettiin paperiprototyyppisiä, jotka piirrettiin käsin.

5.2 Tulokset

Salaisessa liitteessä on listattu tutkimuksen ensimmäisessä vaiheessa kerätty aineisto eli alkuperäiset ilmaukset ja niistä muodostuneet pelkistetyt ilmaukset. Taulukossa 5 on esitetty aineiston pohjalta tehty analyysi. Pelkistettyjä ilmauksia tuli yhteensä 29 kappaletta ja ne jaettiin kahdeksaan eri alaluokkaan toimintoihin liittyvien järjestelmän näkymien perusteella. Näitä alaluokkia ovat Tuotteen tiedot, Tuotelista, Kontaktin tiedot, Kontakttilista, Kaupan tiedot, Tarjouksen tiedot, Raportit sekä Muu. Muu-luokan alle laitettiin kaikki ilmaukset, jotka eivät liittyneet yhteen tiettyyn näkymään järjestelmässä.

Taulukko 5: Tulokset tutkimuksen ensimmäisestä vaiheesta.

Pelkistetyt ilmaukset (M = Muokkaava / T = Tarkistava)	Tuotteen tiedot	Tuotelista	Kontaktin tiedot	Kontakttilista	Kaupan tiedot	Tarjouksen tiedot	Raportit	Muu
Tuotteen hintojen tarkistaminen (T)	X							
Tuotteen varatuksi merkitseminen (M)	X							
Yleisesti tietojen tarkistaminen (T)								X
Tuotelistan käyttö (T)		X						
Kaupan tietojen tarkistaminen (T)					X			
Myyntissä olevien tuotteiden tarkistaminen (T)		X						
Kontaktin luominen (M)					X			
Tuotteen myyntitilanteen tarkistaminen (T)	X							
Kaupan vakuutustietojen selvittäminen (T)					X			
Kauppasopimuksen tekeminen (M)					X			
Kaikki toiminnot (M, T)								X

Tuotelistan käyttö (T)		X						
Kontaktitietojen tarkistaminen (T)			X					
Tuotteen varaaminen (M)	X							
Tuotteen tietojen tarkistaminen (T)	X							
Kaikki toiminnot (M, T)								X
Tuotteen varaaminen (M)	X							
Kontaktin tietojen hakeminen (T)				X				
Kaupan tietojen tarkistaminen (T)					X			
Myyntiraporttien lukeminen (T)							X	
Tuotelistan käyttö (T)		X						
Tuotelista ja tuotteen tietojen hakeminen (T)	X	X						
Kontaktin tietojen hakeminen (T)				X				
Tuotteen varaaminen (M)	X							
Tarjouksen tietojen hakeminen (T)						X		
Uuden kontaktin luominen (M)			X					
Tuotteen tietojen hakeminen ja tuotteen varaaminen (M, T)	X							
Tarjouksien tietojen hakeminen (T)						X		
Tietojen hakeminen yleisesti (T)								X
Yhteensä (kpl): (M: 10, T: 22)	9	5	2	2	5	2	1	4

Taulukosta 5 nähdään, että Tuotteen tiedot oli useimmin mainittu näkymä. Toiseksi eniten mainittiin Tuotelista sekä Kaupan tiedot. Tuotelista valittiin toiseksi näkymäksi Kaupan tietojen sijaan, koska Tuotelista-näkymä on pääasiallinen reitti Tuotteen tiedot -näkymään. Näiden lisäksi päätettiin tehdä prototyyppi Tunnistenumerohaku-näkymästä, koska se on toinen yleinen tapa navigoida Tuotteen tiedot -näkymään ja mahdollistaa useamman Tuotelistaan ja Tuotteen tietoihin liittyvän käyttötapauksen suorittamisen. Tuloksista nähdään lisäksi, että tiedon selvittäminen koettiin mobiilikäytössä tärkeämmäksi tarkoitukseksi tiedon muokkaamiseen verrattuna.

Järjestelmän mobiilikäyttö koettiin tarpeelliseksi. Osa käyttäjistä käyttää nykyistä käyttöliittymää aktiivisesti mobiililaitteilla, mutta suurin osa oli sitä mieltä, että nykyinen käyttöliittymä ei ole erityisen joustava mobiilikäyttöä varten. Alla on nostettu esimerkkejä tutkimuksen aineistosta (Salainen liite), joista tarve mobiilikäytölle ja ilmenee.

”Ylipäättään tommonen light versio CRM:stä olis tosi mahtava et se kulkis puhelimessa mukana.” Myyjä.

”Mä oon tehny kauppasoppareitakin puhelimella... Tosi paljon tulee käytettyä puhelimen kautta.” Myyntipäällikkö.

Osa käyttäjistä olisi kiinnostunut käyttämään järjestelmää mobiililaitteilla, mutta käytössä olevan käyttöliittymän käyttäjäkokemus mobiililaitteilla oli liian huono tätä tarkoitusta varten. Alla on tähän liittyvä nosto aineistosta.

”Mä käytän tällä hetkellä tosi vähän mobiilina tätä, just sen takia että se ei oo mikään joustava siihen. Et se kestä kauan ja hankala käyttää...” Aluepäällikkö.

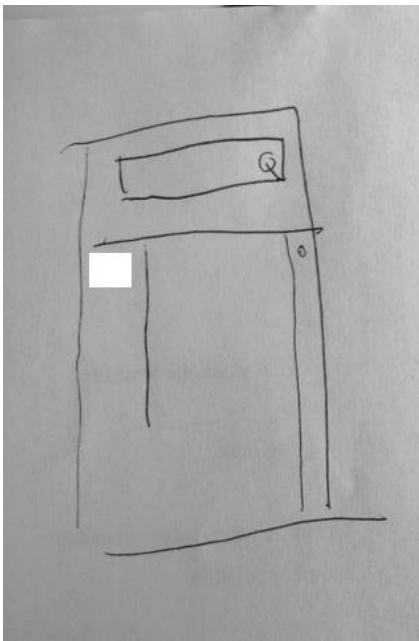
Luonnostelu

Luonnosteluvaiheessa suurin osa haastateltavista mainitsi käytössä olevan käyttöliittymän mobiilikäytön ongelmaksi jatkuvan käyttöliittymän lähentämisen (zoom) tarpeen, koska tietoa on liian paljon näkyvillä käyttöliittymässä. Tämä johtuu pääasiassa käyttöliittymän taulukoista, joissa on usein todella paljon sarakkeita: esimerkiksi Tuotelistalla sarakkeita on yhteensä 14. Nämä eivät mahdu älypuhelimien näytölle, mikä aiheuttaa hankaluuksia halutun tiedon löytämisessä.

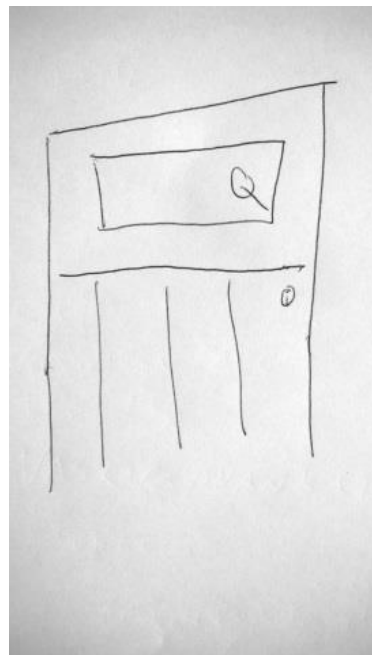
”iPadissa käyttö onnistuu ihan hyvin, mutta kännykällä on vaikeempi kun pitää zoomailla isoissa taulukoissa.” Myyntipäällikkö.

”iPadilla tulee käytettyä kotona. Kännykällä harvemmin kun on vaan niin julmetun pieni se näyttö tälle järjestelmälle, että ei pysty käyttämään.” Myyjä.

Valittujen näkymien mobiilikäyttöliittymän luonnoksissa ja niihin liittyvissä ideoissa selkeitä yhteisiä piirteitä olivat yksinkertainen näkymä, jonka yläosassa on hakukenttä ja hakutuloksissa hyvin rajattu määrä sarakkeita. Kuvissa 9 ja 10 on kaksi tutkimustilanteessa tehtyä luonnosta mobiilikäyttöliittymälle.



Kuva 9: Tutkimuksen ensimmäisessä vaiheessa tehty luonnos mobiilikäyttöliittymästä



Kuva 10: Vaihtoehtoinen luonnos.

Yksi luonnostelussa hyväksi koettu suunnitteluidea tiedon määrän rajoittamiseksi oli, että taulukossa näytettäisiin vain rajattu määrä sarakkeita ja viimeisessä sarakkeessa olisi jonkinlainen lisätietonappi. Ideana oli, että lisätietonappia painamalla esiin tulisi niin sanottu lisätietodialogi, joka sisältäisi kyseisen rivin loput sarakkeet jonkinlaisessa listassa. Dialogissa voisi olla myös nappi, josta pystyy varaamaan tuotteen yhdellä painalluksella. Lisätietonappia, josta dialogi aukeaa on luonnosteltu kuvissa 9 ja 10 taulukon oikeanpuolimmaisessa sarakkeessa.

Luonnoksien pohjalta toteutettiin matalan tason prototyyppi analyysin perusteella valituista näkymistä. Lopullinen matalan tason prototyyppi on esitetty salaisessa liitteessä.

6. VAIHE 2 – PROTOTYYPIN ITEROINTI

Tutkimuksen toisessa vaiheessa oli tarkoitus testata matalan tason prototyyppiä käyttäjillä ja pyrkiä testaamaan suunnitteluideat ennen korkean tason prototyypin toteutusta. Testauksen perusteella matalan tason prototyyppiä iteroitiin tarvittaessa vastaamaan paremmin käyttäjien tarpeita ja sen pohjalta luotiin korkean tason prototyyppi.

6.1 Menetelmä

Tutkimuksen toisessa vaiheessa toteutettiin yksinkertainen käyttäjätestaus matalan tason prototyypille hyödyntäen ääneen ajattelua ja havainnointia tutkimusmenetelminä. Menetelmässä osallistujat suorittavat annetut tehtävät ajattelemalla ääneen, mitä tekevät ja prototyypin käyttöä havainnoitiin. Tällöin käytettävyysoongelmat on helppo havaita ja esille nousevat huomiot kirjataan ylös. Yhtä matalan tason prototyypin iterointikierrosta varten tarvitaan yleisesti ottaen 2-4 henkilöä [5, s. 409]. Tätä prototyyppihaastattelua varten rekrytoitiin yhteensä 3 henkilöä (Taulukko 6), jotka olivat kaikki myyjiä. Muiden käyttäjäryhmien osallistumista ei pidetty tarpeellisena, koska myyjät ovat suurin käyttäjäryhmä ja heidän tarpeet ovat ensisijaiset järjestelmän käytössä.

Taulukko 6: Toisen vaiheen osallistujat.

#	Käyttäjäryhmä
1	Myyjä
2	Myyjä
3	Myyjä

Tutkimusta varten valittiin kolme tehtävää, jotka on listattu alla. Tehtävät valittiin siten, että niissä testattiin sekä tiedon tarkistamista, että muokkaamista käyttäen niitä näkymiä, jotka olivat valikoituneet hyödyllisimmiksi tutkimuksen ensimmäisen vaiheen analyysin perusteella. Oikeat tunnistenumerot on korvattu tähdillä.

1. Selvitä hinta tuotteelle, jonka tunnistenumero on *****.
2. Varaa tuote, jonka tunnistenumero on *****.
3. Muuta toimipistettä tuotteelle, jonka tunnistenumero on *****.

Tutkimus toteutettiin asiakkaan toimipisteellä osallistujille sopivana ajankohtana. Tutkimustilanne nauhoitettiin käyttäen tietokoneen videokameraa ja mikrofonia ja esille nousseet huomiot kirjattiin ylös. Testaukselle varattiin aikaa noin 30 minuuttia osallistujaa kohti. Matalan tason prototyypin iteroitiin tarvittaessa useampia kierroksia kunnes kriittisiä ongelmakohtia ei enää löytynyt. Kun matalan tason prototyypin käytettävyysongelmat oli havaittu ja suunnitteluideat oli testattu, voitiin edetä korkean tason prototyypin toteuttamiseen.

6.2 Tulokset

Matalan tason prototyypin testauksissa kriittisiä ongelmakohtia ei löytynyt ollenkaan, mutta joitakin huomioita nousi esille. Yksi syy sille, että kriittisiltä ongelmilta vältyttiin voi olla se, että prototyyppi vastasi toimintaperiaatteeltaan hyvin pitkälti olemassa olevaa järjestelmää ja kyseisillä sivuilla tehtävät toiminnot olivat varsin yksinkertaisia. Vaikka testausta pyrittiin pohjustamaan kertomalla tutkimusmenetelmän luonteesta ja tarkoituksesta, yhdellä käyttäjistä oli hankaluuksia hahmottaa testauksen ideaa tai hän tunsi itsensä jostain syystä epävarmaksi prototyypin käytössä.

Merkittävin prototyyppihaastattelussa esille noussut huomio oli se, että Lisätieto-näppäimestä aukeavassa dialogissa voisi näyttää enemmän tuotteen tietoja, kuin mitä sen oli alun perin suunniteltu näyttävän. Ensimmäisen vaiheen perusteella nousseiden ideoiden pohjalta ajatuksena oli, että dialogissa näkyisi vain nykyisen järjestelmän taulukon loput sarakkeet, jotka piilotetaan mobiilinäkymässä. Käyttäjät kuitenkin yrittivät etsiä sieltä muutakin tietoa, kuten tuotteen myyntitilannetta, joka ei ole nykyisin Tuotelista-näkymän taulukossa näkyvissä. Näiden tietojen lisääminen lisätieto-dialogiin on mahdollista ja se on hyvä jatkokehitysidea prototyypille. Tämän toteuttaminen on kuitenkin teknisestä näkökulmasta järkevämpi jättää tämän työn ulkopuolelle.

Toinen huomio oli, että yksi käyttäjistä luuli aluksi tunnistenumerohaun olevan Tuotelista-näkymän sisäinen haku. Prototyypin ei toteutettu Tuotelistan sisäistä hakua, koska näkymää haluttiin yksinkertaistaa ja Tunnistenumerohaku tuo esille samat tiedot. Lisäksi Tunnistenumerohaku hakee muitakin tuotteeseen liittyvät tiedot, kuten kontaktit, kaupat, ostot ja tarjoukset. Tätä ei pidetty vakavana käytettävyysongelmana, koska vaikka käyttäjä luulisi tunnistenumerohaun olevan tuotelistan sisäinen haku, se ei mahdollisen alkuhämmennyksen jälkeen varsinaisesti aiheuta ongelmia järjestelmän käytössä, vaan tehostaa sen käyttöä.

Yksi prototyyppihaastattelussa ilmennyt huomio oli myös, että yksi käyttäjistä ei aluksi hyödyntänyt uutta Lisätieto-näppäintä tuotteen tietojen tarkistamiseen. Sen sijaan hän meni tunnistenumeroa painamalla suoraan Tuotteen tiedot -sivulle, mikä on käytössä olevassa järjestelmässä ainoa tapa siirtyä Tuotelistalta Tuotteen tietoihin. Vaikka itse tehtävän suoritus onnistuikin ongelmitta, tämä ei ollut tarkoituksen mukaista, joten sitä

voisi pitää käytettävyyssongelmana. Käyttäjän kanssa käydyn keskustelun pohjalta Lisätieto-näppäin koettiin kuitenkin luontevaksi, kun sen olemassaolon oli havainnut ja sitä oli kerran kokeillut. Lisäksi muut käyttäjät käyttivät lisätietonäppäintä sujuvasti, joten sen voidaan olettaa olevan toimiva suunnitteluidea.

Yleisellä tasolla prototyypin uudet suunnitteluideat, kuten Lisätieto-näppäin ja uusi varaus toiminto, joita matalan tason prototyypissä testattiin, koettiin toimiviksi ja hyödyllisiksi.

6.3 Prototypointi

Korkean tason prototyyppi toteutettiin paperiprototyypin pohjalta. Koska tutkimuksen tavoitteena on mitata ja kehittää mobiilikäyttöliittymän käyttäjäkokemusta, prototyypin ulkoasua lähdettiin toteuttamaan mobiililähtöisesti eli sisältö optimoitiin mobiililaitteita varten. Tiedon näyttämiseen suuremmilla resoluutioilla ei tehty muutoksia käytössä olevaan järjestelmään verrattuna. Teknisestä näkökulmasta korkealla tasolla tämä tarkoittaa tiedon piilottamista mediakyselyiden avulla, kun ikkunan kokoa pienennetään.

Prototyypin toteutusta varten pystytettiin oma kehitysympäristö, johon asennettiin kopio käytössä olevasta järjestelmästä. Kehitysympäristö toimi siis ”hiekkalaatikkona”, jossa korkean tason prototyyppiä voitiin kehittää erillisenä kokonaisuutena ja sitä voitiin ajaa selaimessa omasta URL-osoitteesta. Jotta prototyypin toteuttamiseen valittu Bootstrap ohjelmistokehys saatiin käyttöön kehitysympäristöön, vaaditut CSS ja JavaScript tiedostot piti sisällyttää mukaan HTML koodiin ennen prototyypin toteuttamisen aloittamista.

Ensimmäinen askel responsiivisen ulkoasun luomiseen korkean tason prototyyppiin oli korvata järjestelmän navigaatio. Käytössä olevan järjestelmän navigaatio sisälsi liian monta linkkiä pienemmällä resoluutiolla näytettäväksi, jos sivuttaissuuntaista vieritystä ei haluta tapahtuvan. Tähän ongelmaan Bootstrap ohjelmistokehys tarjoaa valmiin ratkaisun, jossa navigaatiolinkit piilotetaan valikon alle, kun ikkunan leveys pienenee tietyn pysäytyspisteen (oletuksena 768 pikseliä) alle. Tämä on yleinen responsiivisilla sivuilla nähtävä ratkaisu, josta käytetään nimitystä Kytkinnavigaatio (Toggle navigation) [53].

Navigaation muuttaminen responsiiviseksi vaati koko navigaation rakenteen korvaamisen, koska käytössä oleva rakenne ei vastannut Bootstrap navigaation vaatimaa rakennetta. Bootstrap ohjelmistokehityksen tarjoama kytkinnavigaatio tuki alinavigaatioita, joten käytössä olevan järjestelmän alinavigaatiolinkit voitiin näyttää kätevästi ilman toiminnallisuuden muuttamista. Tämänkaltaisten muutoksien toteuttaminen itse olisi kasvattanut työmäärää merkittävästi.

Navigationin toiminnallisuuden toteuttamisen jälkeen oli luonnollista hahmotella hieman ulkoasua ja sen värimaailmaa, mihin haettiin suuntaviivoja käytössä olevasta järjestelmästä. Käytössä olevan järjestelmän ulkoasu ei kuitenkaan ole erityisen modernin näköinen, joten tämän prototyypin luominen oli hyvä mahdollisuus kokeilla jotain uutta ja hakea ulkoasuun hieman virkistystä. Bootstrap tarjoaa muun muassa tyylikkään ja ilmaisen kokoelman logoja, joita otettiin käyttöön muutamassa kohdassa prototyypissä. Yleinen ulkoasun päivitys ei kuitenkaan ollut tämän työn pääasiallinen keskittymispiste, joten tähän käytettiin aikaa vain rajallisesti.

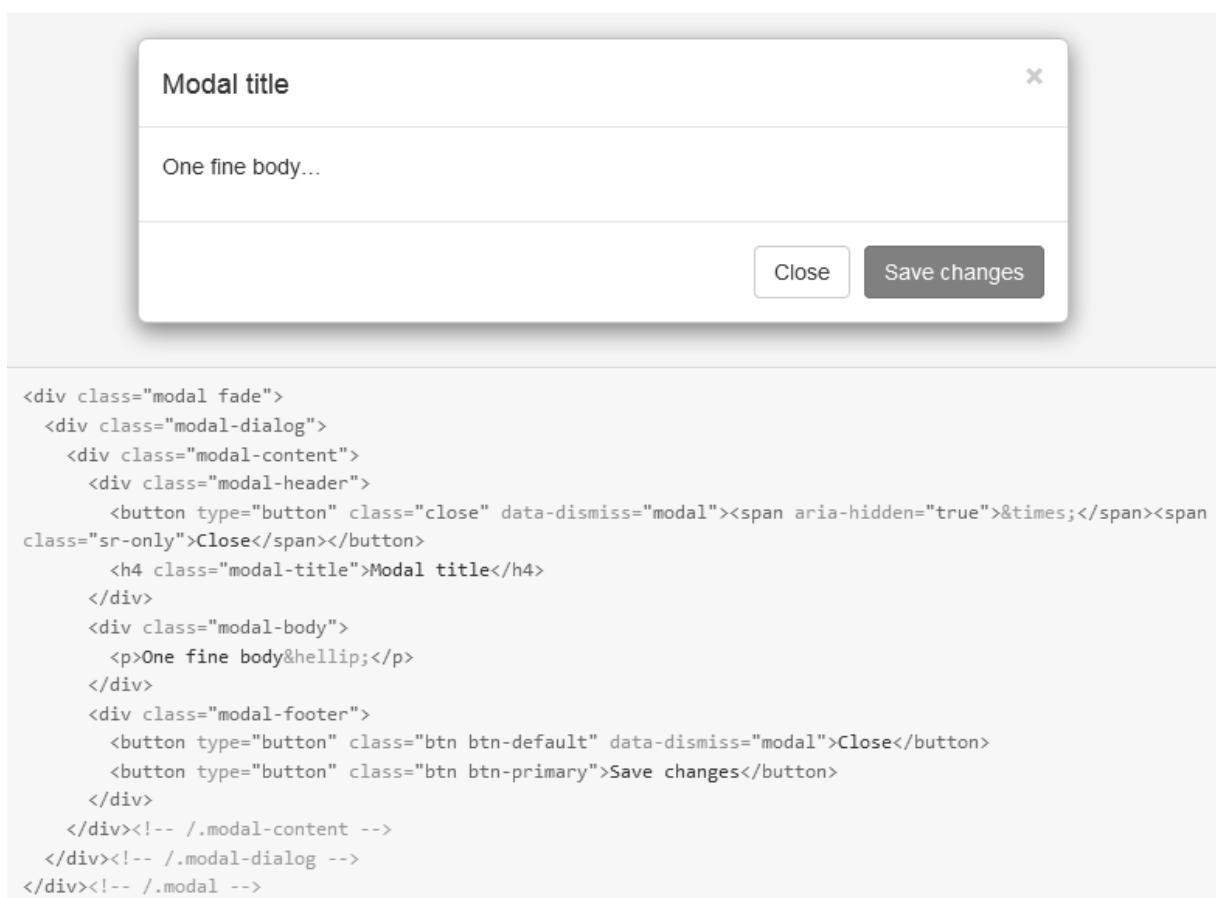
Navigationin ja ulkoasun päivittämisen jälkeen seurasi valittujen näkymien sisällön muuttaminen responsiiviseksi. Tätä hankaloitti se, että käytössä oleva järjestelmä oli tehty hyödyntäen vanhoja HTML-elementtirakenteita, joiden avulla joustavan ruudukkopohjan käyttöönottoaminen on käytännössä mahdotonta. Toisin sanoen ulkoasujen rakenteet oli luotu pääasiassa hyödyntäen `<table>`-elementtejä, joiden sisällä sisältö oli näytetty `<tr>` ja `<td>` -lohkoissa. Jotta Bootstrap ohjelmistokehityksen responsiivinen ruudukkopohja saatiin käyttöön, `<table>`, `<tr>` ja `<td>` -elementit täytyi korvata `<div>`-elementeillä, joille annettiin kehityksen määrittelemät luokat. Tämä oli varsin mekaanista työtä ja koska valitut näkymät sisälsivät paljon sisältöä, siihen kului suuri osa prototyypin toteutukseen kuluneesta ajasta.

Kun näkymien rakenne oli korvattu hyödyntämään joustavaa ruudukkopohjaista järjestelmää, seuraava askel oli tiedon näyttämisen rajaaminen pienemmillä resoluutioilla. Tämä tapahtui piilottamalla elementtejä, kuten taulukon sarakkeita, valikoita ja painikkeita ikkunan leveyden pienentyessä. Tätä varten Bootstrap ohjelmistokehitys tarjoaa valmiit luokat (Taulukko 7), joiden avulla pystyy kätevästi määrittämään mitkä elementit näkyvät milläkin pysäytyspisteillä.

Taulukko 7: Tiedon näyttämisen ja piilottamiseen tarkoitetut apuluokat Bootstrap ohjelmistokehityksessä [55].

	Älypuhelimet ($<768\text{px}$)	Tabletit ($\geq 768\text{px}$)	Pöytäkoneet / Kannettavat ($\geq 992\text{px}$)	Pöytäkoneet ($\geq 1200\text{px}$)
.visible-xs-*	Näky	Piilossa	Piilossa	Piilossa
.visible-sm-*	Piilossa	Näky	Piilossa	Piilossa
.visible-md-*	Piilossa	Piilossa	Näky	Piilossa
.visible-lg-*	Piilossa	Piilossa	Piilossa	Näky
.hidden-xs	Piilossa	Näky	Näky	Näky
.hidden-sm	Näky	Piilossa	Näky	Näky
.hidden-md	Näky	Näky	Piilossa	Näky
.hidden-lg	Näky	Näky	Näky	Piilossa

Samassa yhteydessä toteutettiin tuotelistan mobiilinäkymään kaavailtu uusi Lisätieto-näppäin ja siihen liittyvä logiikka. Tuotelistalta piilotettiin siis ensiksi sarakkeita muutamaa tärkeintä saraketta lukuun ottamatta, käyttäen *hidden-xs* -luokkaa. Tämän jälkeen Lisätieto-näppäin tuli tuotelistan viimeiseen sarakkeeseen ja sille annettiin puolestaan luokka *visible-sm*, koska se näytetään ainoastaan mobiililaitteilla. Lisätietonapista aukeava dialogi toteutettiin hyödyntäen Bootstrap ohjelmistokehyksen tarjoamaa Modal-toiminnallisuutta (Kuva 11). Dialogin sisään haettiin tuotteen tiedot, jotka näkyvät normaalisti tuotelistan sarakkeissa, ja lisättiin painike josta tuotteen pystyi varaamaan.



Kuva 11: Esimerkki Modal-toiminnosta [55].

Prototyyppiin otettiin lisäksi käyttöön jonkin verran muitakin Bootstrap ohjelmistokehyksen tarjoamia luokkia, koska ne tarjosivat paljon hyväksi todettuja tyyllittelyjä valmiina käytettäviksi, mikä nopeutti prototyypin toteutusta. Lopullinen korkean tason prototyyppi on esitetty salaisessa liitteessä.

7. VAIHE 3 – KÄYTETTÄVYYSTESTAUS

Tutkimuksen viimeisessä vaiheessa tehtiin käytettävyystestaus sekä tutkimuksessa kehitetylle korkean tason prototyypille että käytössä olevalle järjestelmälle. Testi suoritettiin molemmilla järjestelmällä, jotta testin avulla kerätty data voidaan muuttaa merkitykselliseksi tiedoksi. Tarkoituksena oli siis verrata testien tuloksia toisiinsa ja mitata kuinka kehitetty prototyyppi palvelee käyttäjien tavoitteita verrattuna käytössä olevaan järjestelmään.

7.1 Menetelmä

Tutkimuksen kolmannessa vaiheessa toteutettiin samat käytettävyyssmittaukset käytössä olevalle järjestelmälle sekä tutkimuksessa kehitetylle korkean tason prototyypille. Testauksien jälkeen keskusteltiin prototyypin käytön herättämisestä ajatuksista ja mielipiteistä lyhyen puolistrukturoidun haastattelun avulla. Testauksia varten rekrytoitiin kahdeksan osallistujaa, jotka olivat samat henkilöt kuin ensimmäisessä vaiheessa yhtä poikkeusta lukuun ottamatta (Taulukko 8). Testauksille varattiin aikaa yksi tunti osallistujaa kohden.

Taulukko 8: Kolmannen vaiheen osallistujat.

#	Käyttäjärühmä
1	Myyjä
2	Myyjä
3	Myyntipäällikkö
4	Myyjä
5	Aluepäällikkö
6	Myyjä
7	Myyjä
8	Myyntipäällikkö
9	Myyjä

Testauksissa tehtyjen kyselyiden vastauksien keräämisessä hyödynnettiin Google Forms -lomaketta, joka tallensi tulokset taulukkoon Google Drive -palvelussa. Käytettävyystestaus muodostui neljästä tehtävästä. Tehtävät valittiin siten, että ne testasivat molempia tutkimuksen ensimmäisessä vaiheessa ilmenneitä mobiilikäytön tarkoituksia: kahdessa tehtävässä testattiin tiedon tarkistamista ja kahdessa tiedon muokkaamista. Valitut tehtävät olivat tehtäviä, joita tutkimuksen ensimmäisen vaiheen

perusteella koettiin tarpeellisiksi mobiilikäytössä ja ne pyrittiin liittämään johonkin todenmukaiseen tilanteeseen.

Kaikki osallistujat suorittivat tehtävät samassa järjestyksessä käyttäen samaa älypuhelinta, jotta kaikilla olisi mahdollisimman yhtenäinen käyttäjäkokemus järjestelmän käytössä. Toinen vaihtoehto olisi ollut antaa käyttäjien käyttää omia älypuhelimiaan tehtävien suoritukseen, jolloin he olisivat olleet tottuneempia käytössä olevaan puhelimeen. Valinta tehtiin kuitenkin sillä perusteella, että prototyypin käyttö vaati pieniä ennakoasetuksia, mikä olisi hankaloittanut testaustilannetta ja käyttöliittymän toimivuudesta kaikilla alustoilla ei ollut täyttä varmuutta. Käytössä oli Nokia Lumia 820 Windows Phone älypuhelin. Testauksissa käytetyt anonymisoidut tehtävät on listattu alla. Alkuperäiset tehtävät on kerrottu salaisessa liitteessä.

1. **Tunnistenumerohaun käyttö:** Asiakas soittaa kun olet liikkeellä. Heti puhelun päätyttyä sinun täytyy selvittää tunnistenumerohaun avulla tuotteen ***** tiedot.
2. **Tuotteen varaaminen:** Soitat takaisin asiakkaalle, mutta keskusteltuanne hän ilmoittaaakin olevansa kiinnostunut eräästä toisesta myynnissä olevasta tuotteesta. Päätät etsiä kyseisen tuotteen tuotelistalta ja varata sen.
3. **Tietojen muokkaaminen:** Myöhemmin neuvottelet vielä asiakkaan kanssa. Neuvottelujen pohjalta päätät muokata saman tuotteen tietoja.
4. **Tuotelistan käyttö:** Asiakas on erityisen kiinnostunut eräästä tuotemerkestä. Haluat selvittää mikä on uusimman myynnissä olevan kyseisen tuotemerkin tuotteen tilanne.

Käytettävyydestestauksessa mitattiin tyytyväisyyttä tehtävätasolla (Task Level Satisfaction) käyttäen jokaisen tehtävän suorituksen jälkeen tehtävän jälkeistä kyselyä (Post-task questionnaire). Tyytyväisyyttä testitasolla (Test Level Satisfaction) mitattiin testin jälkeisellä kyselyllä (Post-test questionnaire), johon vastattiin molempien järjestelmien käytettävyydestien lopuksi. Lisäksi tutkimuksessa mitattiin tehtävien suoritusajat (Task Time).

Suoritusajat mitattiin kannettavan tietokoneen sovelluksella, jonka ajastin käynnistettiin tehtävän aloitushetkellä. Jokainen tehtävän suoritus aloitettiin samalla tavalla siten, että edellisen tehtävän suorituksen jälkeen puhelin otettiin takaisin osallistujalta ja sovellus suljettiin, avattiin uudelleen ja puhelin annettiin takaisin. Ennen puhelimen antamista takaisin osallistujalle annettiin hetki aikaa lukea seuraava tehtävän anto. Tehtävän aloitus laskettiin alkavaksi siitä hetkestä, kun puhelin annettiin takaisin osallistujalle. Osallistujia ohjeistettiin kertomaan, milloin he olivat suorittaneet tehtävän, jolloin suoritusajan mittausta lopetettiin. Osallistujille ei kerrottu, että suoritusajakoja mitattiin.

Ennen tehtävien suorittamista prototyypin avulla osallistujille annettiin hetki aikaa tutustua prototyyppiin. Koska kaikki käyttäjät olivat kokeneita käytössä olevan

järjestelmän käyttäjiä, suoritusajat eivät olisi muuten vertailukelpoisia käytössä olevan järjestelmän suoritusajojen kanssa. Tehtävät suoritettiin lisäksi siten, että joka toinen osallistuja suoritti testin ensiksi prototyypillä ja joka toinen aloitti puolestaan käytössä olevalla järjestelmällä. Tämän järjestelyn tarkoituksena on saada todenmukaisempi käsitys uusien ominaisuuksien vaikutuksesta suoritusaikoihin [45].

Tutkimuksessa käytettäväksi tehtävän jälkeiseksi kyselyksi valikoitui Single Ease Question (SEQ). Se on yksinkertainen kysymys (Kuva 12) johon vastataan asteikolla 1-7 välillä ”Erittäin vaikea” ja ”Erittäin helppo”. Tutkimuksen [46] mukaan yksinkertaisuudestaan huolimatta SEQ korreloi hyvin monimutkaisempienkin kyselyiden kanssa ja on käyttäjien ja tutkimuksen järjestäjien kannalta helppo käsitellä.

Kokonaisuudessaan tämä tehtävä oli: *
 Valitse asteikolta sopivin vaihtoehto
 Merkitse vain yksi soikio.

	1	2	3	4	5	6	7
Erittäin vaikea	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Erittäin helppo							

Kuva 12: Single Ease Question (SEQ).

Testitason kyselyksi valikoitui puolestaan System Usability Scale (SUS), joka on John Brooken vuonna 1986 luoma erittäin paljon käytetty testitason kysely [6, s. 194]. Se koostuu 10 kysymyksestä, joihin vastataan asteikolla 1-5 välillä ”Vahvasti samaa mieltä” ja ”Vahvasti eri mieltä”. Alkuperäinen SUS kysely koostuu sekä positiivisesti että negatiivisesti ilmaistusta kysymyksistä, minkä tarkoituksena on vähentää myönteisyyttä ja liioiteltuja vastauksia. Tutkimuksessa [47] havaittiin kuitenkin, että versio, jossa on pelkästään positiivisesti ilmaistuja kysymyksiä, saavuttaa käytännössä identtiset tulokset alkuperäiseen nähden ja että on todella vähän todellista näyttöä kysymysten negatiivisuuden ja positiivisuuden vaihtelemisen tuomista eduista. Pelkkien positiivisesti ilmaistujen kysymysten käyttäminen helpottaa lisäksi tuloksien analysoimista sekä vähentää käyttäjien tekemien virheiden määrää [47]. Tässä tutkimuksessa päädyttiin käyttämään pelkästään positiivisesti ilmaistuja kysymyksiä SUS kyselyssä. Tutkimuksessa käytetty kysely on esitetty kokonaisuudessaan liitteessä A.

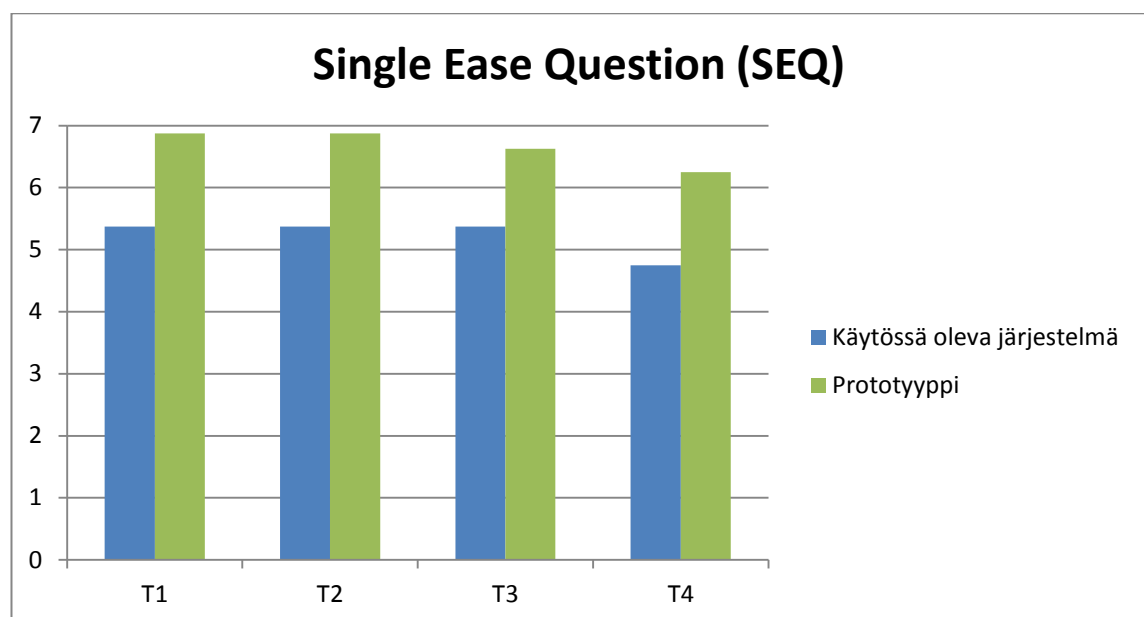
Käytettävyytestien jälkeen pidetyn teemahaastattelun rakenne on esitetty alla.

- 1. Kuvaile avoimesti minkälainen mielikuva sinulle jäi prototyypin käytöstä?**
- 2. Mitkä ovat merkittävimmät erot prototyypin ja nykyisen järjestelmän käytössä mobiililaitteilla?**
- 3. Tuntuiko, että prototyyppi toisi lisäarvoa sinun CRM käyttöön? Miksi?**

Haastattelussa käydään kuitenkin ensiksi läpi testeissä ilmenneitä ongelmakohtia ja pyritään selvittämään niihin liittyvät syyt. Tämän jälkeen haastattelussa edetään teemahaastattelulle tyypillisesti siten, että yllä mainituista kysymyksistä keskustellaan haastateltavan kanssa avoimesti.

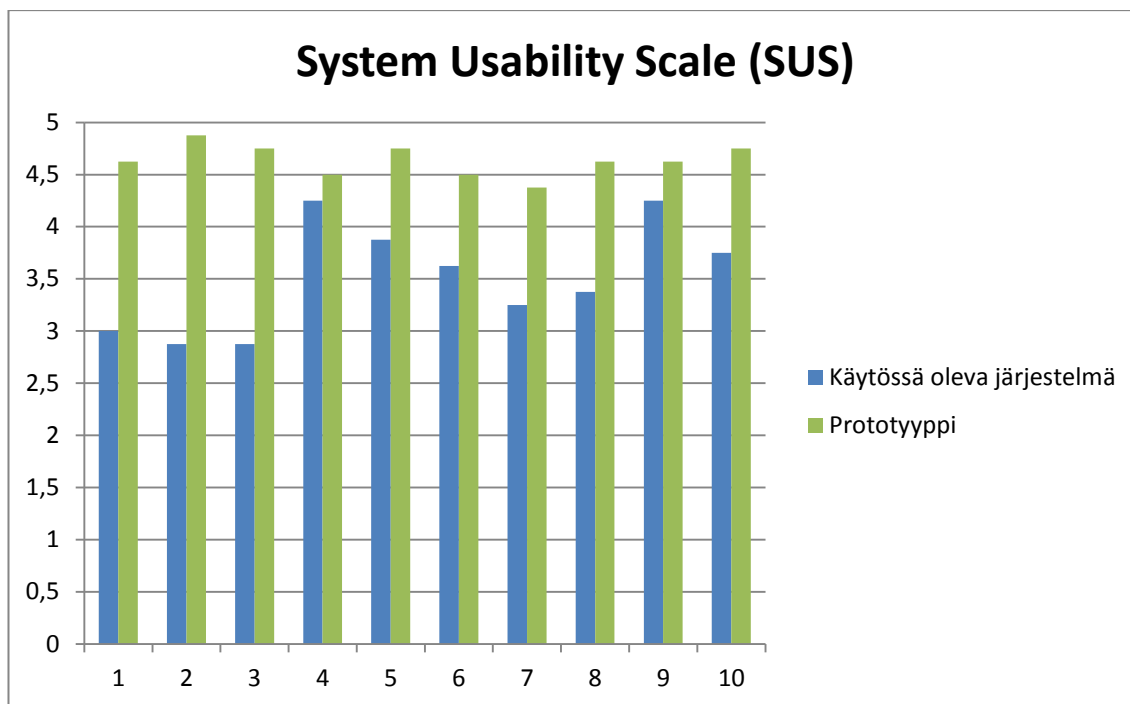
7.2 Tulokset

Tutkimuksen kolmannen vaiheen käytettävyydestien tulokset on esitetty tässä kappaleessa. Vaikka tulokset eivät ole pienen otannan takia tilastollisesti tarkkoja, niiden avulla voidaan vertailla järjestelmien käytettävyyttä suhteessa toisiinsa. Käytössä olevan järjestelmän SEQ tuloksien (Kuvaaja 1) kokonaiskeskiarvo oli 5.2/7 ja prototyypin 6.7/7.



Kuvaaja 1: Single Ease Question (SEQ) kyselyn tuloksien tehtäväkohtaiset keskiarvot prototyypille ja käytössäolevalle järjestelmälle mobiililaitetta käytettäessä.

Käytössä olevan järjestelmän testitason tyytyväisyyttä mittaavan SUS kyselyn tuloksien keskiarvo oli kokonaisuudessaan 3.5/5 ja prototyypin 4.6/5. SUS kyselyn kysymyskohtaiset keskiarvot on esitetty kuvaajassa 2.



Kuvaaja 2: System Usability Scale (SUS) kyselyn tuloksien kysymyskohtaiset keskiarvot prototyypille ja käytössäolevalle järjestelmälle.

On myös syytä huomioida, että puolet osallistujista testasi järjestelmiä eri järjestyksessä kuin toinen puoli. Järjestyksellä oli pieni vaikutus prototyypin eduksi, joka voi johtua siitä, että osallistujat suhteuttavat toisena testattavan järjestelmän vastauksia jonkin verran ensimmäiseen.

Pienestä otannasta johtuen SEQ ja SUS kyselyjen tuloksien tilastollinen merkittävyys on kyseenalainen. Kyselyjen tilastollista merkittävyyttä on kuitenkin arvioitu Wilcoxonin merkittyjen sijalukujen testillä (Wilcoxon signed rank test), jonka avulla lasketut p-arvot on esitetty taulukossa 9. SEQ-kyselyissä tilastollisen merkityksen ($p < 0.05$) saavutti tehtävä 3 ja SUS-kyselyissä puolestaan kysymykset 1, 2, 3, 5 ja 10. Vaikka suuri osa kyselyjen tuloksista ei saavuttanut tilastollista merkitystä, keskiarvoista kuitenkin nähdään, että prototyypin vastaanotto on ollut myönteistä käytössä olevaan järjestelmään verrattuna.

Taulukko 9: Wilcoxonin sijalukujen testillä lasketut p-arvot kyselyjen tuloksille.

	P-arvo
SEQ: Tehtävä 1	0.066
SEQ: Tehtävä 2	0.066
SEQ: Tehtävä 3	0.026
SEQ: Tehtävä 4	0.078
SUS: Kysymys 1	0.026
SUS: Kysymys 2	0.011
SUS: Kysymys 3	0.017
SUS: Kysymys 4	0.581
SUS: Kysymys 5	0.038
SUS: Kysymys 6	0.102
SUS: Kysymys 7	0.088
SUS: Kysymys 8	0.088
SUS: Kysymys 9	0.257
SUS: Kysymys 10	0.039

Liitteessä B on esitetty mitatut suoritusajat sekunneissa. Taulukossa 10 on esitetty tehtäväkohtaiset geometriset keskiarvot suoritusajoille yhden desimaalin tarkkuudella ja 95 % luottamusvälillä lasketut virhemarginaalit. Lisäksi taulukkoon on laskettu geometristen keskiarvojen aritmeettiset keskiarvot.

Taulukko 10: Tehtävien suoritusaikojen keskiarvot ja prosentuaaliset erot prototyypin ja käytössä olevan järjestelmän välillä.

	Käytössä oleva järjestelmä (s)	Prototyyppi (s)
T1: Tunnistenumerohaun käyttö	56.8 (42.5, 75.9)	44.7 (27.0, 73.8)
T2: Tuotteen varaaminen	37.2 (26.2, 52.8)	21.3 (14.3, 31.6)
T3: Tietojen muokkaaminen	42.4 (37.1, 48.5)	41.6 (26.2, 66.1)
T4: Tuotelistan käyttö	70.6 (49.2, 101.2)	69.7 (53.2, 91.3)
Keskiarvo	51.8	44.3

Suoritusaikojen keskiarvo on hieman pienempi prototyypillä käytössä olevaan järjestelmään verrattuna. Virhemarginaalit olivat kuitenkin suuret pienestä näytekoosta johtuen. Tuloksista nähdään, että virhemarginaalit menevät päällekkäin prototyypin ja käytössä olevan järjestelmän välillä, joten tilastollisesti merkittävää eroa suoritusajoissa ei ole.

Huomion arvoista suoritusajoissa on tehtävän kaksi suoritusaika, jonka suoritusaikojen keskiarvo on lähes puolet (43 %) pienempi prototyypillä käytössä olevan järjestelmän keskiarvoon verrattuna. Kyseisessä tehtävässä tehtävänä oli varata tuote, mikä oli tutkimuksen ensimmäisessä vaiheessa eniten esille noussut yksittäinen mobiililaitteilla

hyödylliseksi koettu tehtävä. Tehtävän kolme suoritusaika on puolestaan keskiarvoltaan heikompi prototyypillä, koska käytössä oleva järjestelmä tarjosi tehtävän suoritusta helpottavan oikotien, jota prototyypin ei oltu vielä toteutettu. Kaikki osallistujat eivät tosin hyödyntäneet tätä oikotietä.

Käytettävyydestien jälkeen keskusteltiin kehitysideoista tehtävissä ilmenneiden käytettävyyssongelmien pohjalta selvittämällä näihin liittyviä syitä. Tehtävä neljä (Tuotelistan käyttö) oli osallistujille haasteellisin suorittaa ja siihen liittyen heräsi jonkin verran keskustelua testien jälkeen. Kyseisessä tehtävässä täytyi ensiksi etsiä halutun tuotteen tunnistenumero tuotelistalta ja näppäillä se tunnistenumorohakuun. Koska hakukenttä on sivun ylälaudassa ja haettava tuote noin keskivaiheilla listaa eikä tunnistenumeroa pystynyt älypuhelimella helposti kopioimaan, tunnistenumero pääsi usein unohtumaan tai se täytyi muusta syystä käydä tarkistamassa uudestaan jotta haku saatiin suoritettua onnistuneesti. Käyttyjen keskustelujen perusteella monien osallistujien mielestä olisi hyvä, jos listalla olisi linkki suoraan kyseisen tuotteen tunnistenumorohaun hakutuloksiin. Työpöytäversiossa tämä linkki ei ole yhtä tarpeellinen, koska tunnistenumeron kopioiminen leikepöydälle on helppoa. Siksi lisätietodialogi voisi olla sopiva paikka kyseiselle linkille. Toinen mainittava käytettävyyssongelma on tuotelistan lisätietonäppäin, johon osalla käyttäjistä oli vaikea osua peukalolla. Näppäintä voitaisiin pyrkiä suurentamaan tai sen osuma-aluetta voisi yrittää laajentaa.

Prototyyppi sai tuloksien perusteella osallistujilta myönteisen vastaanoton. Tämä välittyi myös käytettävyydestien jälkeen käydyistä teemahaastatteluista, joista on nostettu esille esimerkkejä eri kysymyksiin liittyen.

1. *"Hyvä kuva jäi, että kun joutuu kuitenkin mobiilisti hommaa käyttää ni huomattavasti nopeuttaa ja helpottaa et ei tarvii sit zoomailla ja kikkailla muuta."* Myyntipäällikkö
1. *"Mielelläni käyttäisin sitä omassa puhelimessanikin."* Myyjä
2. *"Prototyyppi toimii paremmin kännykän näytöllä. Onhan se vanha järjestelmä vähän tutumpi niin siellä on helppo zoomailla mut tuntu tosi helpolta toi uus."* Myyjä
2. *"Mun mielestä se yhteensopivuus nimenomaan sen mobiililaitteen kanssa, että ei tarvii zoomailla sun muuta. Et paljon yksinkertasempi käyttää."* Myyjä
3. *"Melkein päivittäin tulee tilanteita et tarvii CRM:stä jotain kattoo tien päällä tai kotona, et kyl se tulis helpottaa ihan huomattavasti sitä hommaa."* Myyntipäällikkö
3. *"Ite tulis ainakin käytettyä... et pystyis nopesti aina kattoo jonkun homman siitä ilman sitä tietokonettakin."* Myyjä

8. JOHTOPÄÄTÖKSET

Laitemarkkinoiden nopean kehityksen ansiosta mobiililaitteiden huomioimisesta on tullut ajankohtainen ja entistä tärkeämpi aihe järjestelmien suunnittelussa ja kehityksessä. Mobiililaitteiden huomioimiselle on olemassa useita lähestymistapoja, joiden hyödyistä toisiinsa verrattuna käydään parhaillaan aktiivista keskustelua. Korkeimmalla tasolla mobiililaitteiden huomioimisen lähestymistavaksi web-järjestelmissä katsotaan yleisesti olevan kaksi vaihtoehtoa: erillisen mobiilisovelluksen taikka responsiivisen käyttöliittymän toteuttaminen. Mobiilisovelluksia voidaan lisäksi tehdä usealla eri tavalla, joista yleisimmin listataan natiivisovellukset, mobiilit HTML5 web-sovellukset sekä hybridisovellukset.

Kaikilla lähestymistavoilla on omat hyvät ja huonot puolensa. Natiivisovelluksien avulla voidaan saavuttaa paras käyttäjäkokemus, mutta ne ovat kalliita, koska kaikille käyttöjärjestelmille täytyy toteuttaa ja ylläpitää omaa sovellusta. Mobiilit web-sovellukset ovat halpoja toteuttaa, mutta eivät vastaa käyttäjäkokemukseltaan natiiveja sovelluksia johtuen muun muassa mobiiliselaimien puutteellisesta pääsystä laiterajapintaan. Hybridisovellukset ovat puolestaan web-sovelluksia, jotka on paketoitu natiivisovelluksiksi. Niiden avulla voidaan tehdä alustariippumattomia sovelluksia, joilla on pääsy laiterajapintaan, mutta niiden toteuttaminen on suhteellisen kallista puhtaisiin web-sovelluksiin verrattuna.

Responsiivinen web-suunnittelu on mobiililaitteiden huomioimisen lähestymistapa, joka mahdollistaa web-järjestelmän käyttöliittymän toteuttamisen siten, että se mukautuu käytössä olevan laitteen näytön kokoon. Tällä lähestymistavalla on merkittäviä etuja, kuten yhden koodipohjan mahdollistama parempi ylläpidettävyyys ja kustannustehokkuus muihin vaihtoehtoihin verrattuna. Responsiivisen web-suunnittelun avulla saavutetaan yhtenäinen ulkoasu kaikille laitteille ja alustoille, mutta sen haasteena on tällä hetkellä suorituskyky.

Tässä työssä kehitettävän järjestelmän mobiililaitteiden huomioimisen lähestymistavaksi valikoitui responsiivinen web-suunnittelu. Tärkeimpinä syinä valinnalle oli ylläpidettävyyden helppous ja kustannustehokkuus pitkällä tähtäimellä. Järjestelmän pääasiallinen käyttö tapahtuu työpöytätietokoneilla, mikä olisi tehnyt erillisen mobiilisovelluksen toteuttamisesta ja ylläpitämisestä vähemmän kannattavaa yhden koodipohjan ylläpitoon verrattuna.

Yleisesti ottaen responsiivisessa web-suunnittelussa suurin haaste on suorituskyvyn optimointi, koska erityisesti mobiililaitteilla suorituskyky vaikuttaa merkittävästi

käyttäjäkokemukseen. Pelkän käyttöliittymän korvaaminen responsiivisella käyttöliittymällä ei paranna sivuston teknistä suorituskykyä, vaikka responsiivisuuden mahdollistama käytettävyyden parantuminen voi parantaa itse käytön tehokkuutta ja mielekkyyttä. Käyttöliittymän päivityksen lisäksi responsiivisessa web-suunnittelussa täytyy siis miettiä järjestelmän arkkitehtuuria siten, että mobiililaitteille haetaan vain tarvittava tieto sivunlatauksien yhteydessä ja suuremmille resoluutioille ladattavaa tiedon määrää voidaan kasvattaa.

Jos kehitettävän järjestelmän käyttö tapahtuu pääasiassa mobiililaitteilla tai projektissa on hyvät rahalliset ja ajalliset edellytykset mobiilikäyttäjäkokemuksen optimoimiseen, erillisen mobiilisovelluksen toteuttaminen voi olla järkevä lähestymistapa. Mobiilisovelluksissa suurin osa käytöstä tehdään yhä natiivisovelluksissa, koska niiden avulla voidaan saavuttaa paras käyttäjäkokemus. Jos HTML5 standardin kehitys kuitenkin jatkuu ja mobiilien HTML5 web-sovelluksien pääsy laiterajapintaan paranee, niiden avulla voitaisiin saavuttaa lähes natiivisovelluksia vastaava käyttäjäkokemus merkittävästi pienemmillä kustannuksilla. Tällöin on mahdollista, että HTML5-pohjaiset sovellukset voivat nousta vallitsevaksi teknologiaksi mobiilisovelluksien toteuttamisessa.

Työn empiriaosuudessa toteutettiin prototyyppi käytössä olevasta järjestelmästä hyödyntäen käyttäjäkeskeisiä suunnittelumenetelmiä. Työssä toteutettiin kolmivaiheinen käyttäjätutkimus, jonka ensimmäisessä vaiheessa tutkittiin käyttäjien tarpeita hyödyntäen Contextual Inquiry -menetelmää. Ensimmäisessä vaiheessa tarkoituksena oli selvittää, mitä tarpeita järjestelmän mobiilikäyttöön liittyy sekä mitä järjestelmän toimintoja mobiililaitteilla käytetään eniten. Tutkimukseen osallistui kahdeksan osallistujaa ja siinä kävi ilmi, että tiedon tarkistaminen on pääasiallinen tarve järjestelmän mobiilikäytölle. Lisäksi tutkimuksessa valittiin kolme mobiilikäyttöä ajatellen tärkeintä näkymää toteutettavaksi prototyyppiin. Tutkimuksen ensimmäisen vaiheen jälkeen toteutettiin matalan tason prototyyppi mobiilikäyttöliittymästä hyödyntäen paperiprototypointia.

Tutkimuksen toisessa vaiheessa testattiin matalan tason prototyyppiä kolmella käyttäjällä hyödyntäen ääneen ajattelua ja havainnointia tutkimusmenetelminä. Tarkoituksena oli testata suunnitteluideoiden toimivuutta ja löytää mahdolliset käytettävyysongelmat ennen käyttöliittymän varsinaista toteutusta. Testauksien perusteella löydettiin muutama hyödyllinen havainto, mutta pääasiassa suunnitteluideat osoittautuivat toimiviksi, joten niiden pohjalta voitiin luoda korkean tason prototyyppi. Korkean tason prototyyppi luotiin hyödyntäen Twitter Bootstrap ohjelmistokehystä.

Tutkimuksen kolmannessa vaiheessa suoritettiin identtinen käyttäjätutkimus käyttäen korkean tason prototyyppiä sekä käytössä olevaa järjestelmää mobiililaitteella. Tarkoituksena oli mitata molempien järjestelmien käytettävyyttä hyödyntäen tehtävän jälkeisiä kyselyitä, testin jälkeisiä kyselyitä ja suoritusaikojen mittausta, ja verrata

järjestelmien saamia tuloksia toisiinsa. Kolmanteen vaiheeseen osallistui kahdeksan henkilöä. Tuloksien perusteella korkean tason prototyyppi sai hyvän vastaanoton käyttäjiltä.

Kolmannen vaiheen tuloksien tilastollista merkityksellisyyttä laskettiin hyödyntäen Wilcoxonin merkittyjen sijalukujen testiä. Vaikka tutkimukseen osallistui vain kahdeksan osallistujaa, prototyyppi sai tilastollisesti merkittävästi paremmat tulokset puolessa kyselyjen kysymyksistä käytössä olevaan järjestelmään verrattuna, mitä voi pitää hyvänä tuloksena. On todennäköistä, että suuremmalla otannalla prototyyppi olisi saavuttanut vielä suuremmalla osalla kysymyksistä tilastollisesti merkittävästi paremmat tulokset käytössä olevaan järjestelmään verrattuna. Suoritusajojen eroja ei voida pitää tilastollisesti merkittävänä pienestä otannasta johtuen, vaikka niissäkin prototyyppi sai paremmat tulokset.

Tutkimuksen tutkimusmenetelmien valinnat olivat pääosin onnistuneita. Tutkimuksen merkityksellisyyttä olisi kuitenkin voitu parantaa pyrkimällä rekrytoimaan enemmän osallistujia tutkimuksen kolmanteen vaiheeseen. Lisäksi vaikka ensimmäisessä vaiheessa saatiin paljon hyödyllistä tietoa järjestelmän mobiilikäytöstä, mistä oli hyötyä prototyypin toteutuksessa, siinä olisi voinut riittää jokin hieman kevyempikin tutkimusmenetelmä saman lopputuloksen saamiseksi tutkimuksessa.

Työssä tehdyssä prototyypissä korvattiin käytössä olevan järjestelmän käyttöliittymä hyödyntäen responsiivista web-suunnittelua, mikä paransi järjestelmän käytettävyyttä mobiililaitteilla merkittävästi. Vaikka prototyypin latausaikoja ei varsinaisesti koettu hitaiksi, prototyypin käyttäjäkokemusta mobiililaitteilla voidaan kuitenkin yhä optimoida merkittävästi panostamalla mobiililähtöiseen arkkitehtuurin suunnitteluun serverin pään koodissa. Näin voidaan välttää turhan tiedon latautuminen kun järjestelmää käytetään mobiililaitteilla, mikä parantaa järjestelmän teknistä suorituskykyä. Jos prototyyppi halutaan ottaa käyttöön koko järjestelmään, on tehtävä vielä suuri määrä työtä lopun järjestelmän muuttamiseksi prototyypin mukaiseen ulkoasuun.

Tätä opinnäytetyötä voidaan käyttää pohjana kyseessä olevan CRM-järjestelmän sekä uusien järjestelmien kehityksessä. Työ antaa paremmat lähtökohdat työmäärien arvioimiselle erityisesti responsiivisen web-suunnittelun lähestymistavalle mobiililaitteiden huomioimisessa. Työ kartoittaa vaihtoehtoiset lähestymistavat ja huomioon otettavat asiat mobiililaitteiden huomioimisessa käytössä olevassa järjestelmässä. Työssä käsitellyt asiat ovat kuitenkin pääosin yleistettävissä myös muihin järjestelmiin ja niiden huomioimista suositellaan erityisesti uusien järjestelmien kehityksessä.

9. LÄHTEET

- [1] C. Abras, D. Maloney-Krichmar & J. Preece, 2004. User-Centered Design. In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications.
- [2] R. Baharuddin, D. Singh & R. Razali, 2013. Usability Dimensions for Mobile Applications-A Review. Research Journal of Applied Sciences, Engineering and Technology, 5(6), pp. 2225-2231.
- [3] M. H. Baturay, M. Birtane, 2013. Responsive web design: A new type of design for web-based instructional content. Procedia-Social and Behavioral Sciences, 106, pp. 2275–2279.
- [4] N. Bevan, I. Curson, 1999. Planning and implementing user-centred design. In CHI'99 Extended Abstracts on Human Factors in Computing Systems, New York: ACM Press, pp. 137-138.
- [5] H. Beyer & K. Holtzblatt, 1998. Contextual Design: Defining Customer-Centered Systems, Morgan Kaufmann Publishers, San Francisco, CA.
- [6] J. Brooke, 1996. SUS: A Quick and Dirty Usability Scale. In: P.W. Jordan, B. Thomas, B.A. Weerdmeester & I.L. McClelland (Eds.), Usability Evaluation in Industry. London: Taylor & Francis.
- [7] P. Cashmore, Why 2013 Is the Year of Responsive Web Design, 2012 [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://mashable.com/2012/12/11/responsive-web-design>
- [8] A. Charland, B. LeRoux 2011. Mobile application development: web vs. native. Communications of ACM, vol. 54(5), pp. 49–53.
- [9] L. Chittaro, 2011. Designing visual user interfaces for mobile applications. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems. New York, USA, pp. 331-332.
- [10] Device APIs Working Group [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.w3.org/2009/dap/>
- [11] ExactTarget, 2014. 2014 Mobile Behavior Report [WWW]. [Viitattu 9.9.2014]. Saatavissa:

<http://www.exacttarget.com/sites/exacttarget/files/deliverables/etmc-2014mobilebehaviorreport.pdf>

- [12] B. S. Gardner, 2011. Responsive Web Design: Enriching the User Experience. *Connectivity and the User Experience*, pp. 13-19.
- [13] Gartner, 2014. Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments to Grow 4.2 Percent in 2014 [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.gartner.com/newsroom/id/2791017>
- [14] Gartner, 2014. Gartner Says Customer Relationship Management Software Market Grew 13.7 Percent in 2013 [WWW]. [Viitattu: 8.7.2014] Saatavissa: <http://www.gartner.com/newsroom/id/2730317?fnl=search&srcId=1-3478922254>
- [15] J. Garofalakis, V. Stefanis, 2008. MokE: a tool for Mobile-ok evaluation of web content. In: *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, ACM press, Beijing, China, pp. 57–64.
- [16] P. Gasston, 2013. *The Modern Web - Multi-Device Web Development with HTML5, CSS3, and JavaScript*. No Starch Press; 1 edition.
- [17] J. Hird, 2011. The fight gets technical: mobile apps vs. mobile sites [WWW]. [Viitattu 9.9.2014]. Saatavissa: <https://econsultancy.com/blog/7832-the-fight-gets-technical-mobile-apps-vs-mobile-sites#i.am3s4o16zfnwsg>
- [18] A. Holzinger, 2005. Usability engineering methods for software developers. *Communication of the ACM* 48, pp. 71–74.
- [19] A. Holzinger, M. Errath, 2007. Mobile computer Web-application design in medicine: some research based guidelines. *Univ Access in the Information Society International Journal*, 6, 1, pp. 31-41.
- [20] K.-Y Huang, 2009. Challenges in Human-Computer Interaction Design for Mobile Devices. In *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA. ACM.
- [21] N. P. Huy & D. vanThanh, 2012. Evaluation of mobile app paradigms. In *Proceedings of the 10th International Conference on Advances in Mobile Computing and Multimedia, MoMM '12*, ACM, pp. 25–30.

- [22] IDC, 2013. Tablet Shipments Forecast to Top Total PC Shipments in the Fourth Quarter of 2013 and Annually by 2015, According to IDC [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.idc.com/getdoc.jsp?containerId=prUS24314413>
- [23] ISO, 1999. 13407 Human-Centred Design Processes for Interactive Systems, ISO 13407: 1999.
- [24] R. D. Johansen, T. C. P. Britto & C. A. Cusin, 2013. CSS browse selector plus: a JavaScript library to support cross-browser responsive design. Proceedings of the 22nd international conference on World Wide Web companion, pp. 27-30.
- [25] T. Jokela, N. Iivari, J. Matero & M. Karukka, 2003. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In Proceedings of the Latin American conference on Human-computer interaction. ACM Press. pp. 53-60.
- [26] T. Kadec, 2013. Implementing Responsive design, New riders, Berkley.
- [27] S. Khalaf, 2014. Apps Solidify Leadership Six Years into the Mobile Revolution [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.flurry.com/bid/109749/Apps-Solidify-Leadership-Six-Years-into-the-Mobile-Revolution#.U-M8uoCSwng>
- [28] M. Korf, E. Oksman, 2012. Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options [WWW]. [viitattu 28.1.2014]. Saatavissa: https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options
- [29] J. Krenz-Kurowska, 2013. Web design trends for 2013 [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://99designs.com/designer-blog/2013/02/21/web-design-trends-for-2013/>
- [30] S. Loveland, 2011. Human Computer Interaction That Reaches Beyond Desktop Applications. In Proceedings of the 42nd ACM technical symposium on Computer science education, New York, NY, USA, pp. 595–600.
- [31] E. Marcotte, 2011. Responsive Web Design, A Book Apart, New York.
- [32] E. Marcotte, 2010. Responsive Web Design [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://alistapart.com/article/responsive-web-design>

- [33] J. Mazzei, 2014. Responsive Web Design. Business NH magazine [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://millyardcommunications.com/index.php?src=news&refno=3426&category=News>
- [34] C. B. Mills, L. J. Weldon, 1986. Reading text from computer screens. *ACM Computing Surveys*, 19, pp. 329-358.
- [35] S. Mohorovičić, 2013. Implementing responsive web design for enhanced web presence. In: 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO), IEEE. pp. 1206-1210.
- [36] M. Nebeling & M. C. Norrie, 2013. Responsive design and development: methods, technologies and current issues. *Web Engineering*, Springer Berlin Heidelberg, pp. 510-513.
- [37] T. Nicholson. Grid. [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://605.wikispaces.com/grid>
- [38] D. Norman, & S. Draper, 1986. *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates.
- [39] S. Pastore, 2012. The Role of Open Web Standards for Website Development Adhering to the One Web Vision. *International Journal of Engineering and Technology*, vol. 2, no. 11, pp. 1824-1834.
- [40] A. Payne & P. Frow, 2005. A Strategic Framework for Customer Relationship Management, *Journal of Marketing*, 69, pp. 167–76.
- [41] J. Peppard, 2000. Customer relationship management in financial services, *European Management Journal*. 18. pp. 3312–327.
- [42] G. Podjarny, 2013. Real World RWD Performance – Take 2 [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.guypo.com/uncategorized/real-world-rwd-performance-take-2/>
- [43] Push Technology, 2014. What Mobile Development Model is Right for You? [WWW]. [Viitattu 9.9.2014]. Saatavissa: http://cdn2.hubspot.net/hub/188121/file-1041703765-pdf/Mobile_App_Dev_White_paper_FINALCOPY.pdf?submissionGuid=b86768d5-ab1c-4b15-9092-e3aa57fe68ec

- [44] J. A. Sanchez, 2012. Infographic: 2013 The Year of Responsive Design [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.uberflip.com/blog/infographic-2013-the-year-of-responsive-design>
- [45] J. Sauro, 2011. Compared to what? Making Sense Of Customer Metrics [WWW]. [Viitattu: 8.7.2014] Saatavissa: <http://www.measuringusability.com/blog/compared-what.php>
- [46] J. Sauro, J. S. Dumas, 2009. Comparison of three one-question, post-task usability questionnaires. Proc. CHI '09. ACM Press. pp. 1599-1608.
- [47] J. Sauro, & J. R. Lewis, 2011. When designing usability questionnaires, does it hurt to be positive? Proceedings of ACM SIGCHI, New York, NY, USA. ACM. pp. 2215-2223.
- [48] N. Serrano, J. Hernantes, & Gallardo, G., 2013. Mobile Web Apps. IEEE Software, vol. 30(5), pp. 22-27.
- [49] D. Stone, C. Jarrett, M. Woodroffe & S. Minocha, 2005. User interface design and evaluation. Morgan Kaufman Publishers. London.
- [50] Telerik, 2012. HTML5 Adoption - Fact or Fiction. [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.telerik.com/whitepapers/kendo-ui/html-adoption-survey-2012>
- [51] Telerik, 2013. The HTML5 vs. Native Debate is Over and the Winner is... [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.telerik.com/whitepapers/kendo-ui/the-html5-vs.-native-debate-is-over-the-winner-is>
- [52] The New Multi-screen World: Understanding Cross-platform Consumer Behaviour, 2012 [WWW]. [Viitattu 27.9.2014]. Saatavissa: http://ssl.gstatic.com/think/docs/the-new-multi-screen-world-study_research-studies.pdf
- [53] Toggle Navigation [WWW]. [Viitattu 27.9.2014]. <http://codepen.io/bradfrost/full/sHvaz>
- [54] J. Tuomi, A. Sarajärvi, 2002. Laadullinen tutkimus ja sisällönanalyysi. Tammi, Helsinki, Finland.
- [55] Twitter Bootstrap [WWW]. [Viitattu 27.9.2014]. <http://getbootstrap.com/>

- [56] K. Vredenburg, J. Y. Mao, P. W. Smith & T. Carey, 2002. A Survey of user-centered design practice. In Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press. pp. 471–478.
- [57] W3C, 2012. Media Queries [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://www.w3.org/TR/css3-mediaqueries/>
- [58] WebmasterFormat, 2012. Media Queries: An Evolution in Web Design [WWW]. [Viitattu 9.9.2014]. Saatavissa: <http://webmasterformat.com/blog/media-queries-and-responsive-web-design>
- [59] A. Williams, 2009. User-centered design, activity-centered design, and goal-directed design: a review of three methods for designing web applications. In: Proceedings of the 27th ACM International Conference on Design of Communication, ACM Press, 2009. pp. 1-8.

12. **6. Mielestäni järjestelmä on johdonmukainen. ***

Valitse asteikolta sopivin vaihtoehto

Merkitse vain yksi soikio.

	1	2	3	4	5	
Täysin eri mieltä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Täysin samaa mieltä

13. **7. Kuvittelen, että useimmat oppisivat järjestelmän käytön erittäin nopeasti. ***

Valitse asteikolta sopivin vaihtoehto

Merkitse vain yksi soikio.

	1	2	3	4	5	
Täysin eri mieltä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Täysin samaa mieltä

14. **8. Mielestäni järjestelmän käyttö oli erittäin intuitiivista (= oli erittäin helppo arvata, miten järjestelmä toimii). ***

Valitse asteikolta sopivin vaihtoehto

Merkitse vain yksi soikio.

	1	2	3	4	5	
Täysin eri mieltä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Täysin samaa mieltä

15. **9. Tunsin oloni itsevarmaksi, kun käytin järjestelmää. ***

Valitse asteikolta sopivin vaihtoehto

Merkitse vain yksi soikio.

	1	2	3	4	5	
Täysin eri mieltä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Täysin samaa mieltä

16. **10. Osaisin käyttää järjestelmää ilman, että minun täytyy opetella mitään uusia asioita. ***

Valitse asteikolta sopivin vaihtoehto

Merkitse vain yksi soikio.

	1	2	3	4	5	
Täysin eri mieltä	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Täysin samaa mieltä

Palvelun tarjoaa



Google Forms

LIITE B: MITATUT SUORITUSAJAT

Käytössä oleva järjestelmä (s)		Prototyyppi (s)
Tehtävä 1		
Käyttäjä #1	30	25
Käyttäjä #2	70	140
Käyttäjä #3	75	65
Käyttäjä #4	85	50
Käyttäjä #5	53	33
Käyttäjä #6	71	60
Käyttäjä #7	50	26
Käyttäjä #8	43	27
Tehtävä 2		
Käyttäjä #1	55	15
Käyttäjä #2	25	10
Käyttäjä #3	40	20
Käyttäjä #4	63	35
Käyttäjä #5	25	15
Käyttäjä #6	45	40
Käyttäjä #7	47	22
Käyttäjä #8	20	30
Tehtävä 3		
Käyttäjä #1	45	35
Käyttäjä #2	45	55
Käyttäjä #3	40	130
Käyttäjä #4	40	37
Käyttäjä #5	48	22
Käyttäjä #6	50	50
Käyttäjä #7	45	35
Käyttäjä #8	30	25
Tehtävä 4		
Käyttäjä #1	55	55
Käyttäjä #2	65	70
Käyttäjä #3	180	108
Käyttäjä #4	55	101
Käyttäjä #5	55	81
Käyttäjä #6	70	65
Käyttäjä #7	94	40
Käyttäjä #8	48	63